The Computational Power of Biochemistry

Luca Cardelli
Microsoft Research
with
Gianluigi Zavattaro
University of Bologna

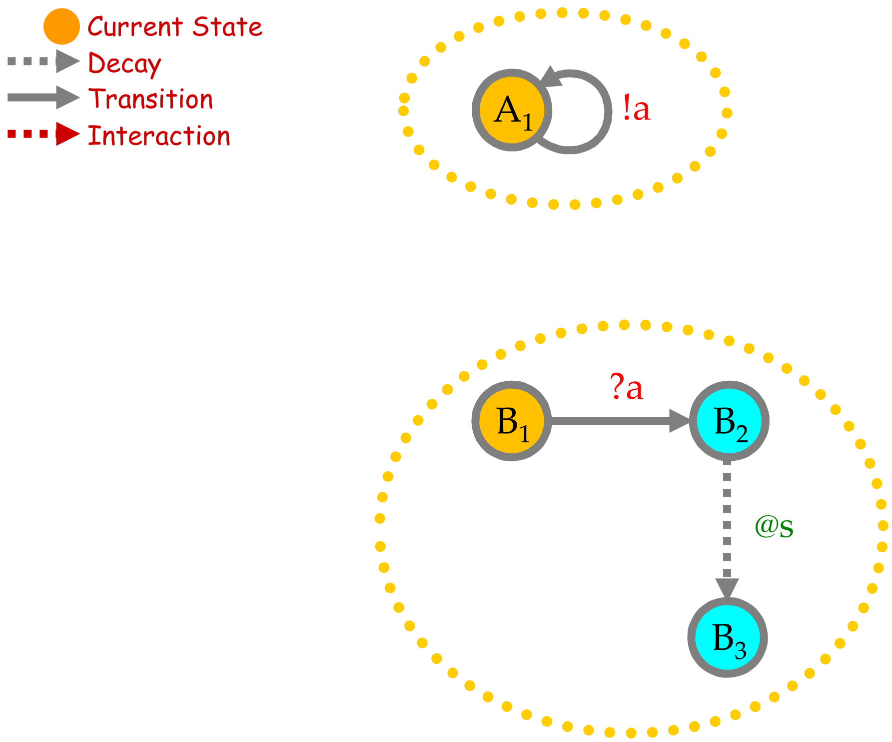
2008-06-05, Formal Methods in Systems Biology, Cambridge
http://LucaCardelli.name

(Macro-) Molecules as (Interacting) Automata

• Concurrent	(math is based on processes, not functions)
• Asynchronous	(no global clock)
• Stochastic	(or nondeterministic)
• Stateful	(e.g. phosphorylation state)
• Discrete	(transitions between states)
• Interacting	(an “interaction” can be pretty much anything you want that changes molecular state)

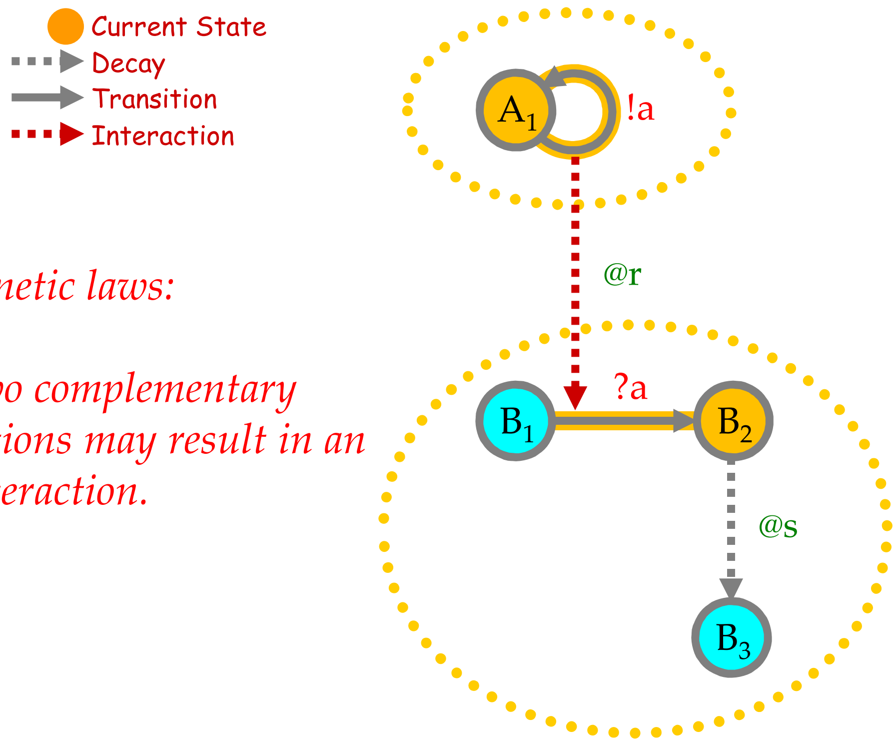
- Based on work on process algebra and biological modeling; see references in related papers.

Interacting Automata



A_1	is a *state*
a	is a *channel* i.e. a named *interaction interface* (e.g. a surface patch)
?,!	indicate any *complementarity* of interaction (e.g. charge)
?a, !a	indicate *complementary actions,*
@r, @s	are rates

Interacting Automata

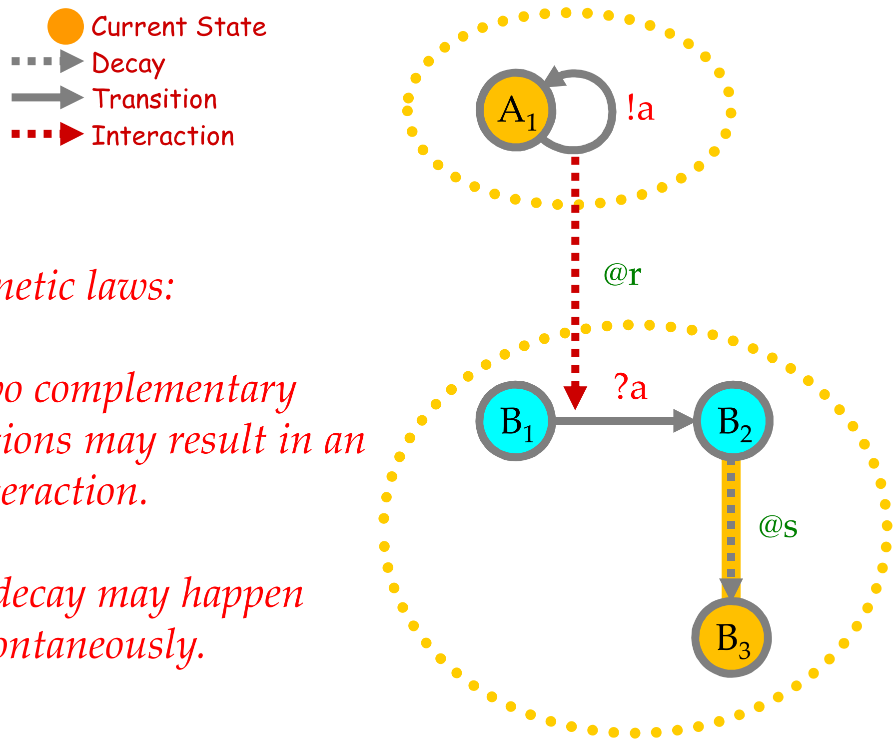


Kinetic laws:

Two complementary actions may result in an interaction.

A_1	is a *state*
a	is a *channel* i.e. a named *interaction interface* (e.g. a surface patch)
?,!	indicate any *complementarity* of interaction (e.g. charge)
?a, !a	indicate *complementary actions*, joined by an interaction arrow
@r, @s	are rates

Interacting Automata



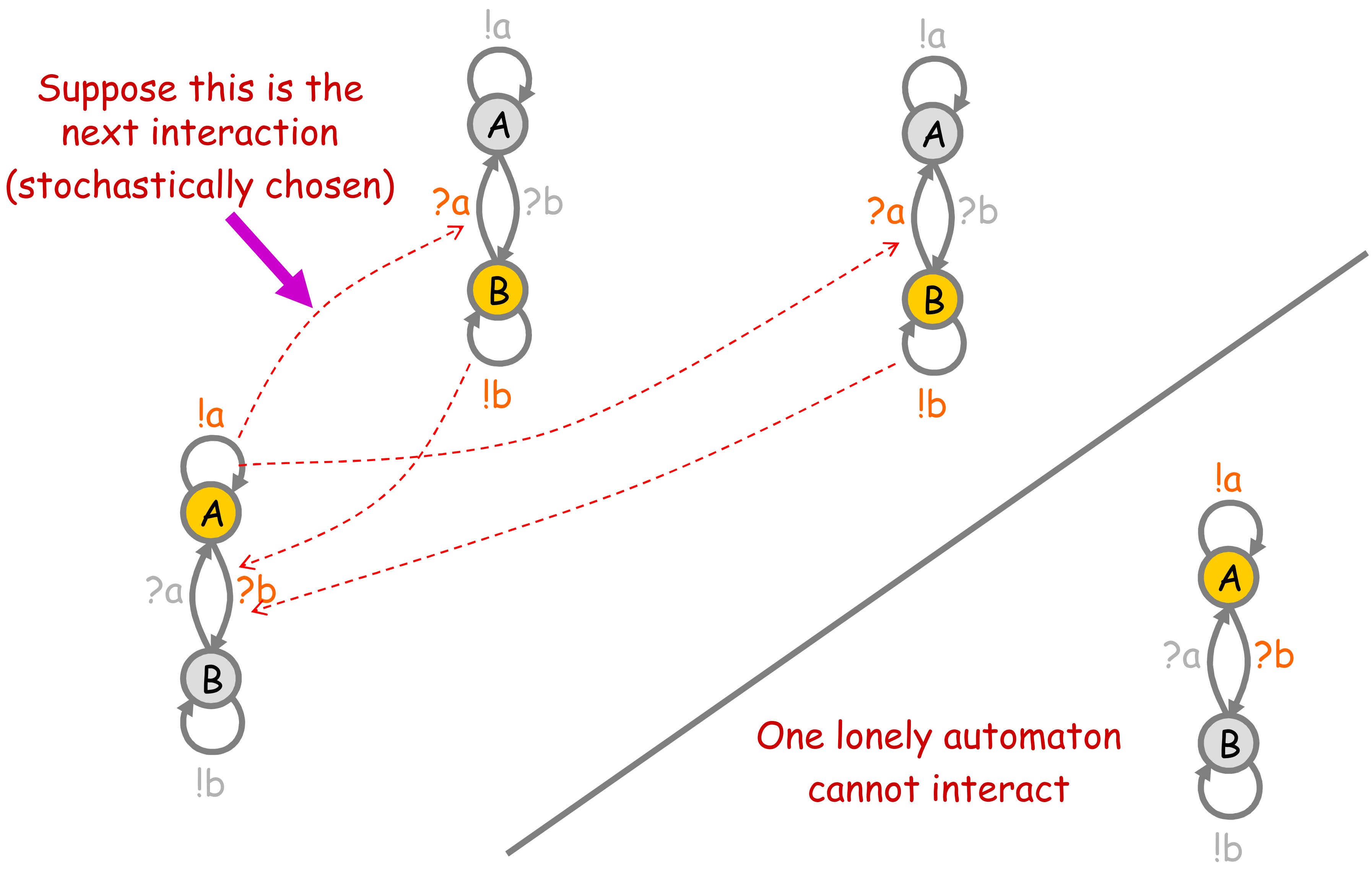
Kinetic laws:

Two complementary actions may result in an interaction.

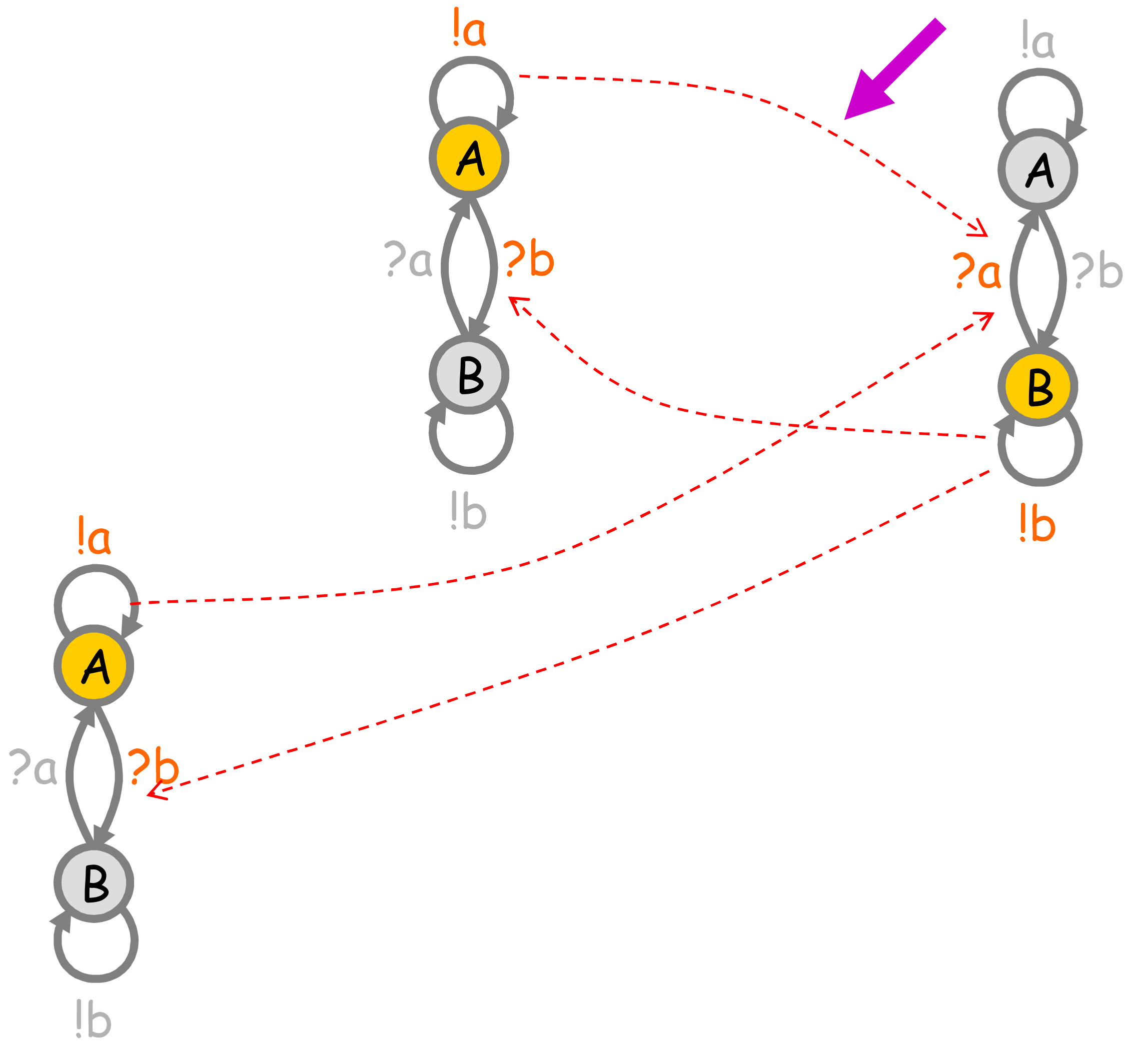
A decay may happen spontaneously.

A_1	is a *state*
a	is a *channel* i.e. a named *interaction interface* (e.g. a surface patch)
?,!	indicate any *complementarity* of interaction (e.g. charge)
?a, !a	indicate *complementary actions*, joined by an interaction arrow
@r, @s	are rates

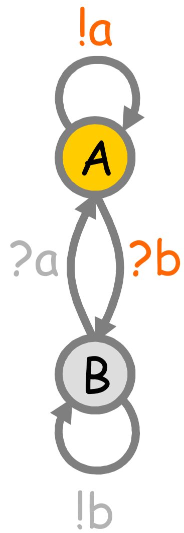
Interactions in a Population

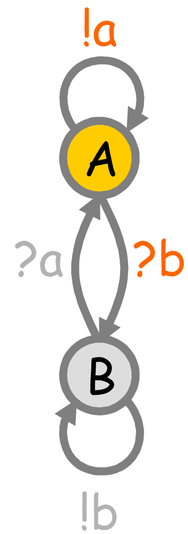


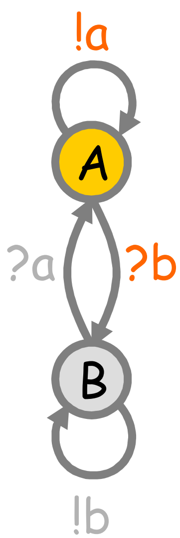
Interactions in a Population



Interactions in a Population

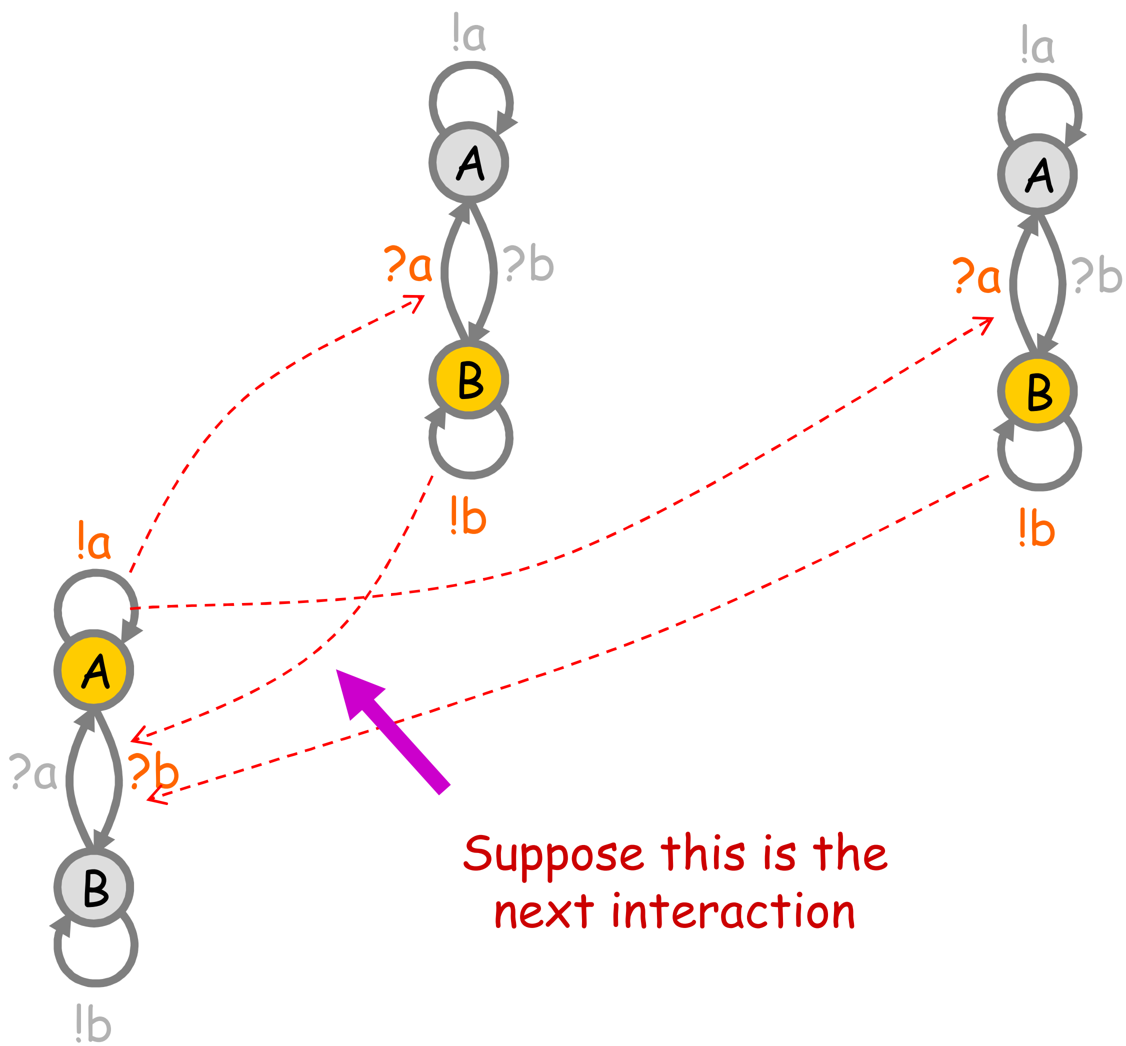




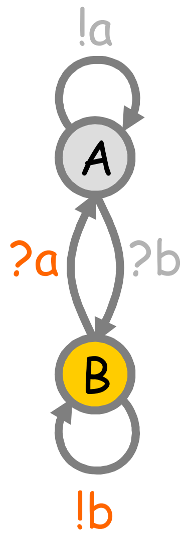


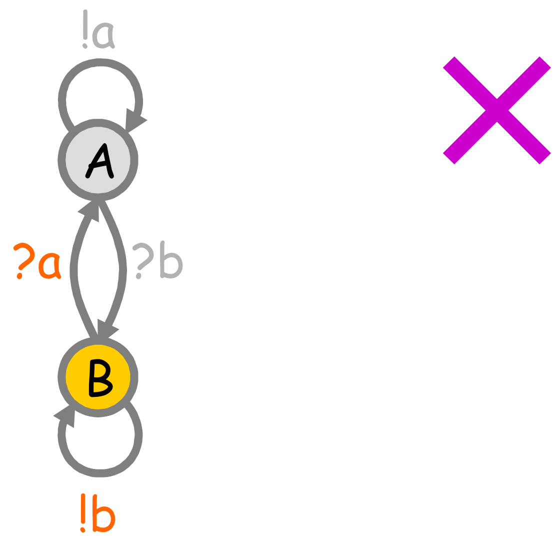
All-A stable population

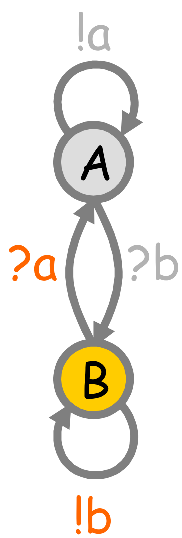
Interactions in a Population (2)



Interactions in a Population (2)



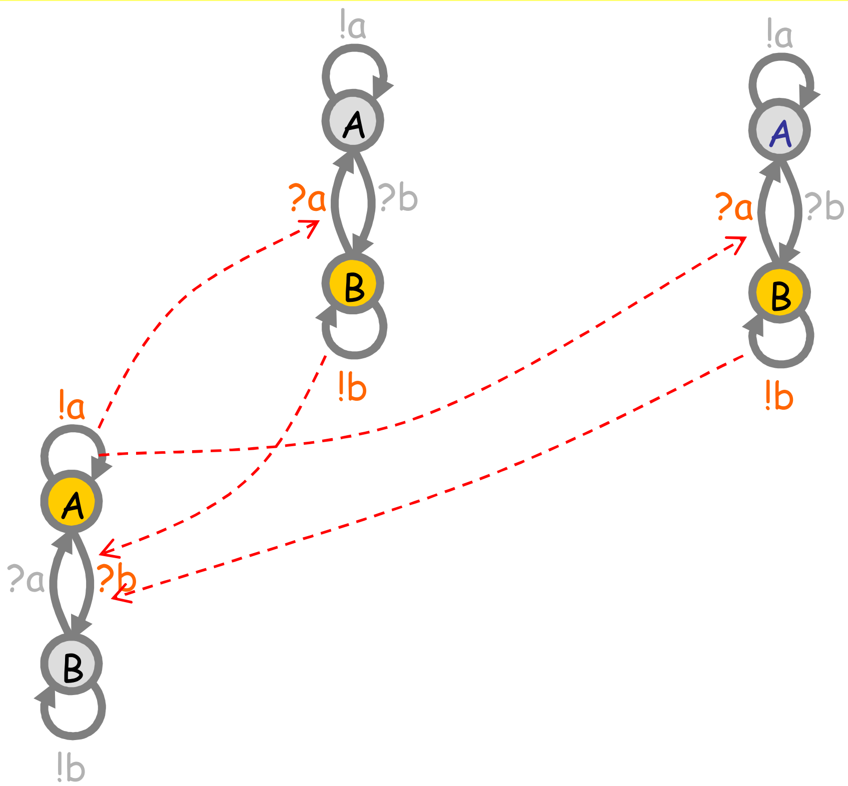


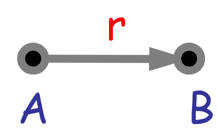


All-B stable population

Nondeterministic population behavior ("multistability")

CTMC Semantics



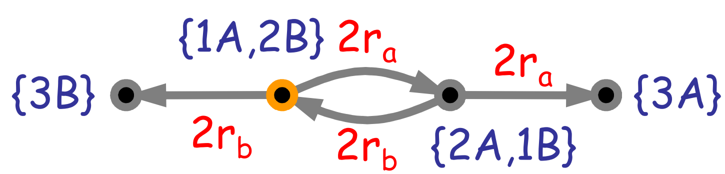


CTMC
(homogeneous) Continuous Time Markov Chain
- directed graph with no self loops
- nodes are system states
- arcs have transition rates

Probability of holding in state A:

$Pr(H_A > t) = e^{-rt}$

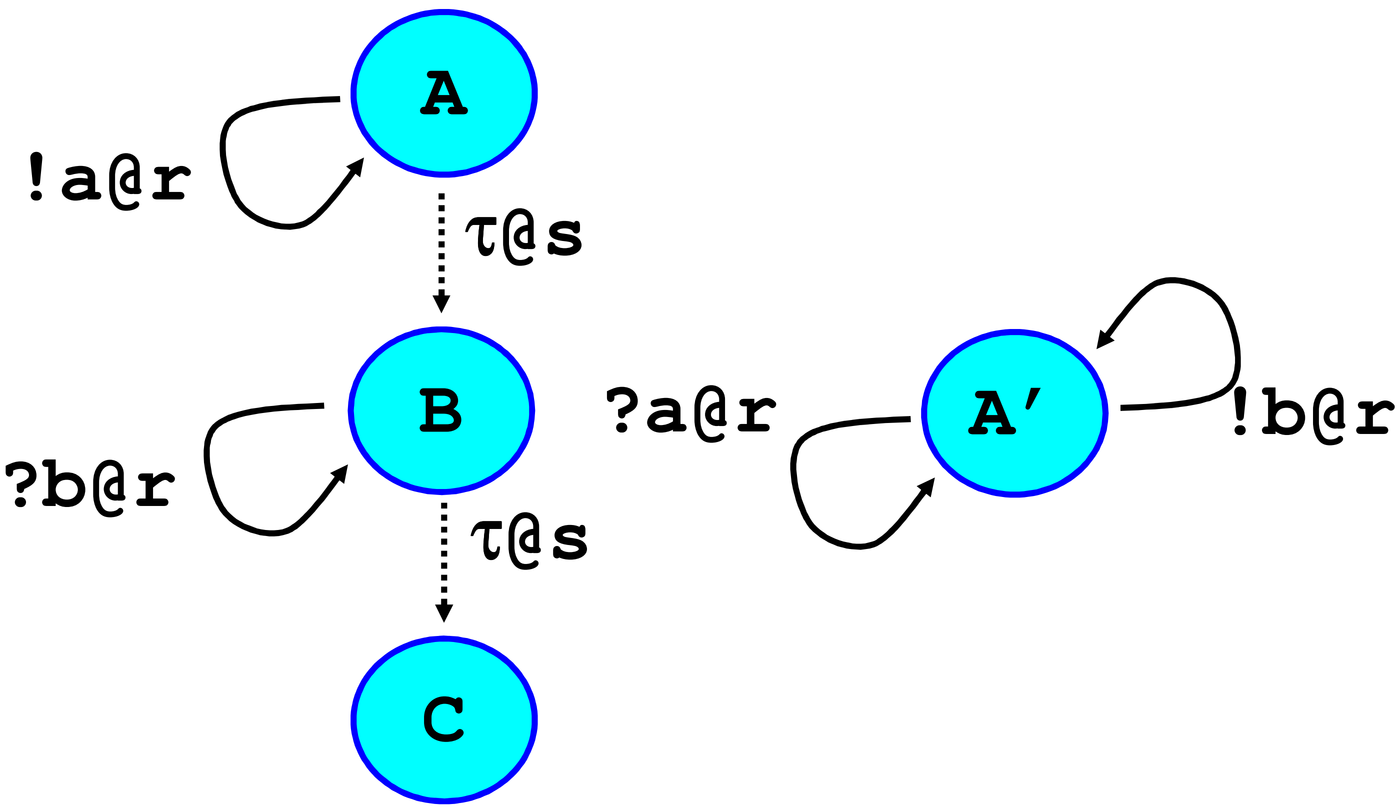
in general, $Pr(H_A > t) = e^{-Rt}$ where R is the sum of all the exit rates from A



CTMC

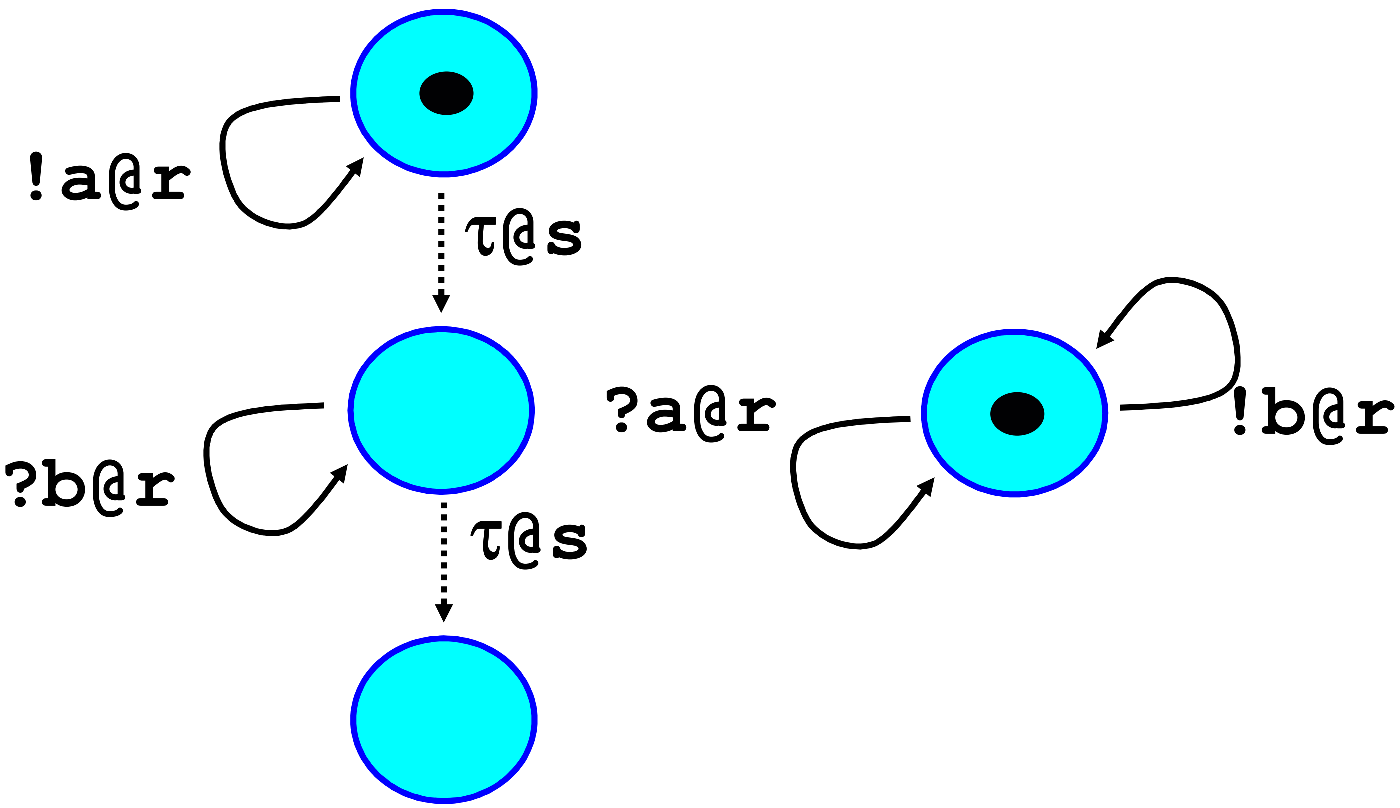
Termination

Example 1: Does it Halt?



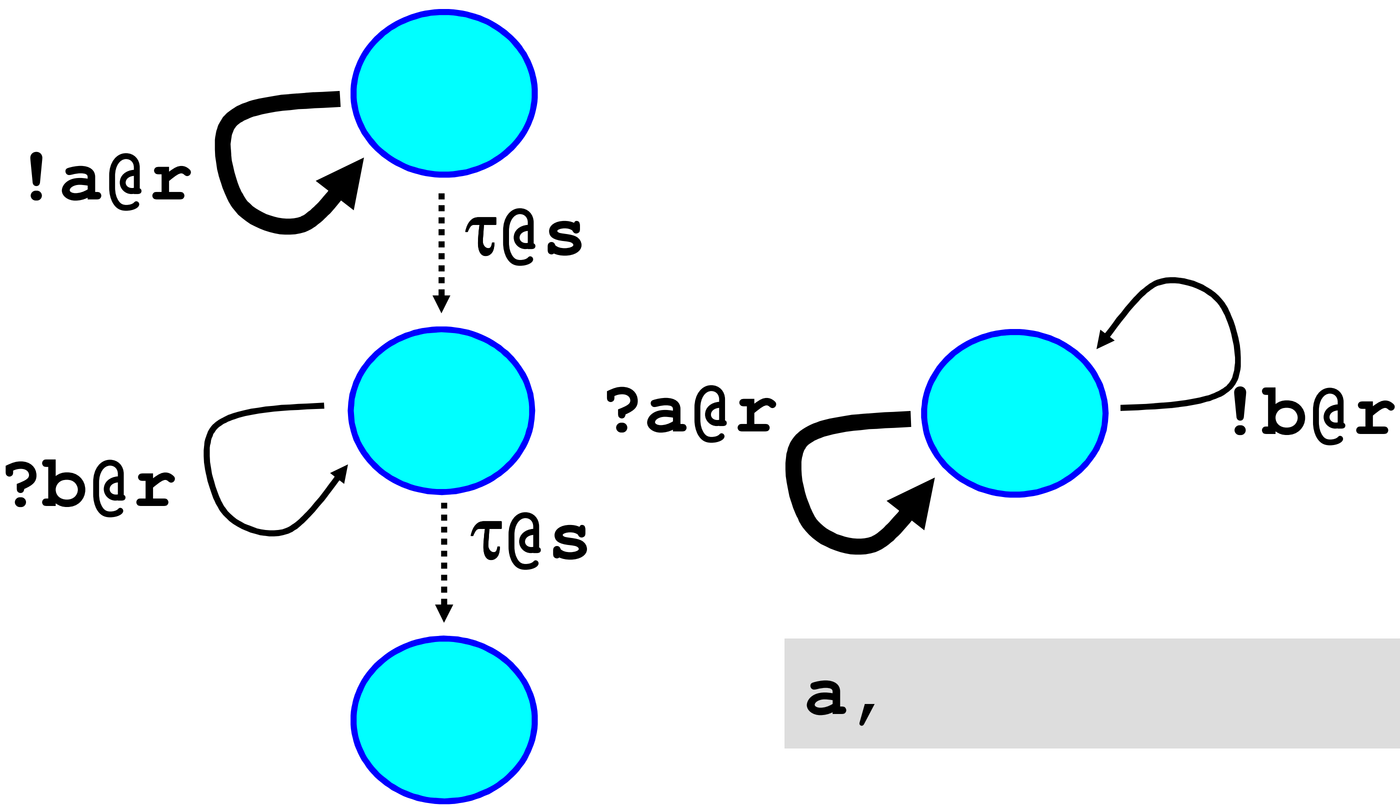
- Starting population: **A|A′**

Example 1: Does it Halt?



- Starting population: **A|A′**

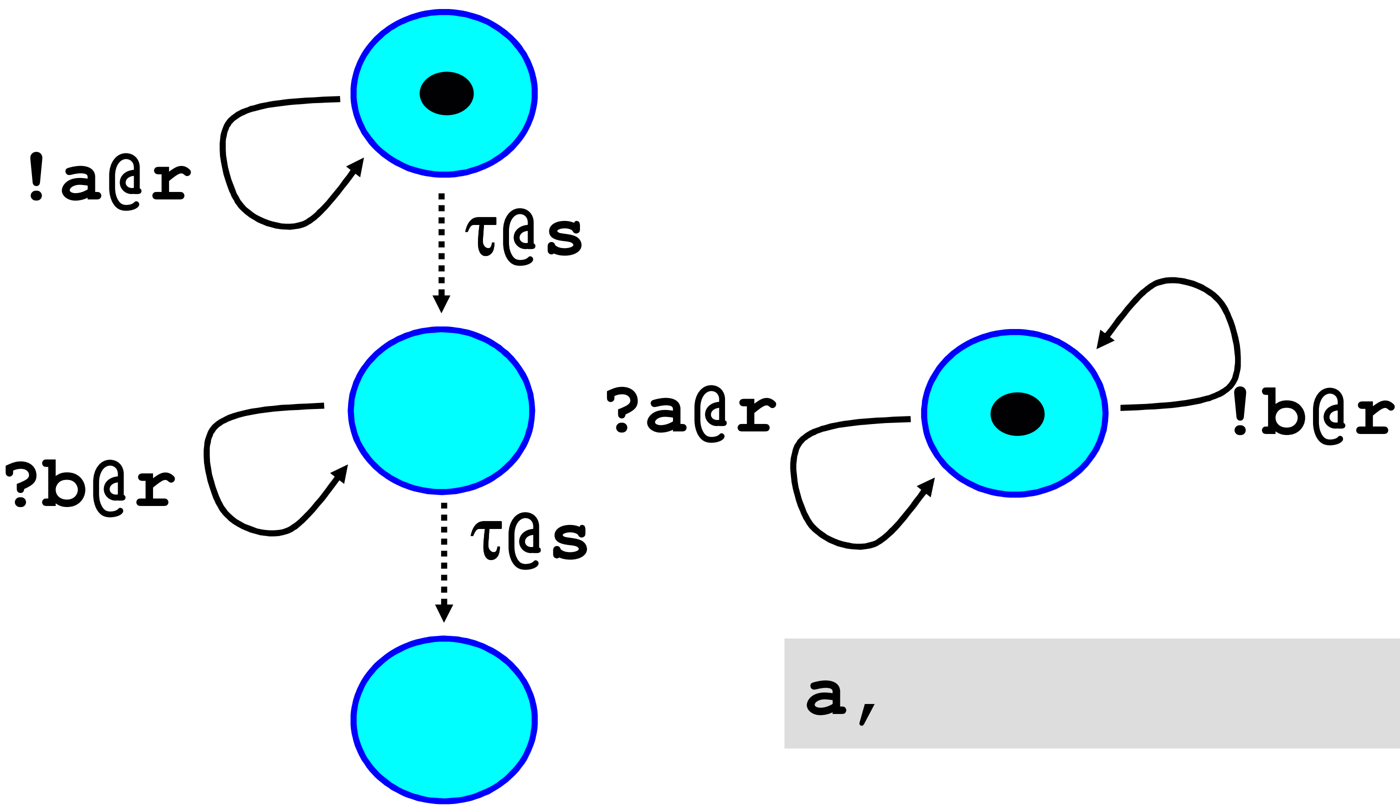
Example 1: Does it Halt?



a,

- Starting population: **A|A′**

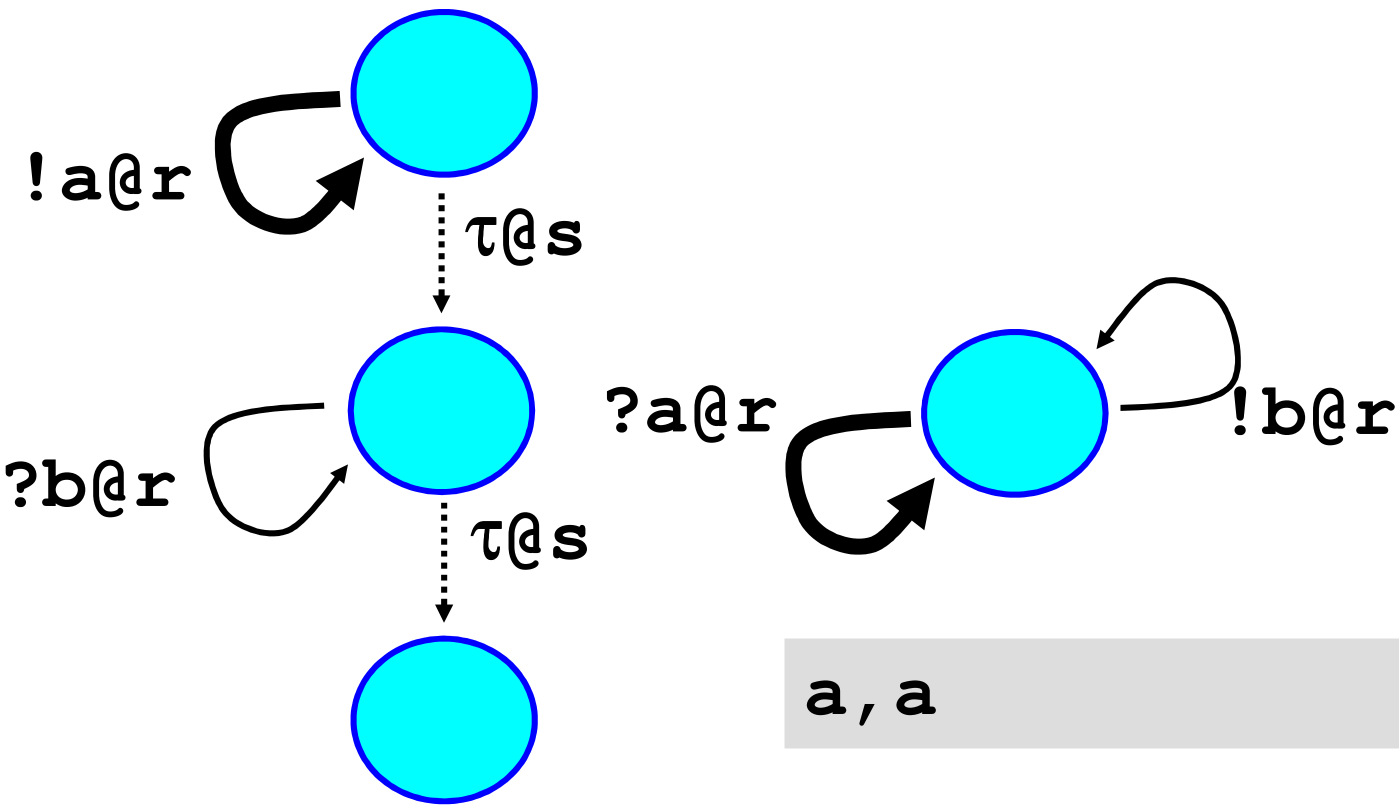
Example 1: Does it Halt?



a,

- Starting population: **A|A′**

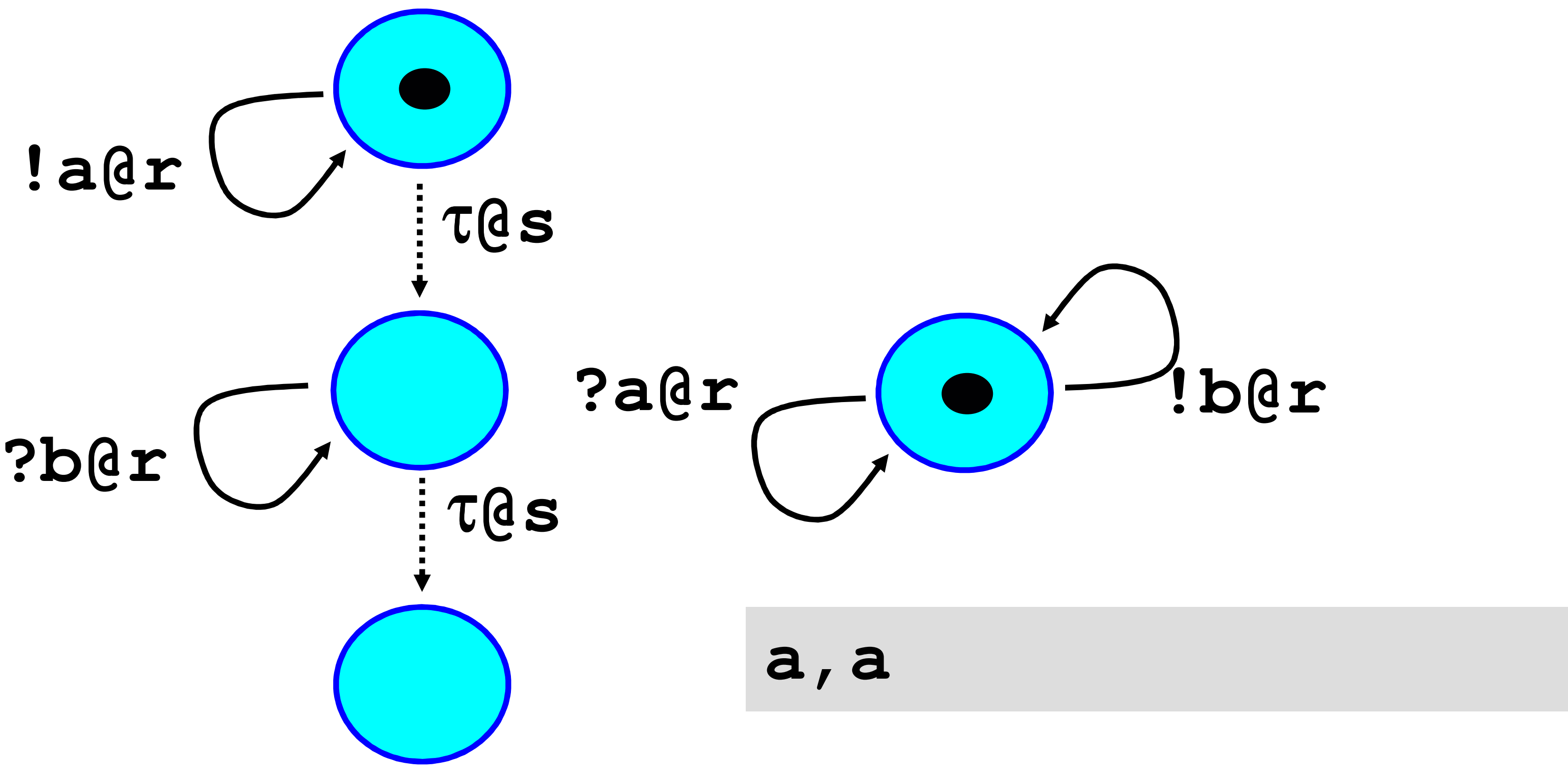
Example 1: Does it Halt?



a,a

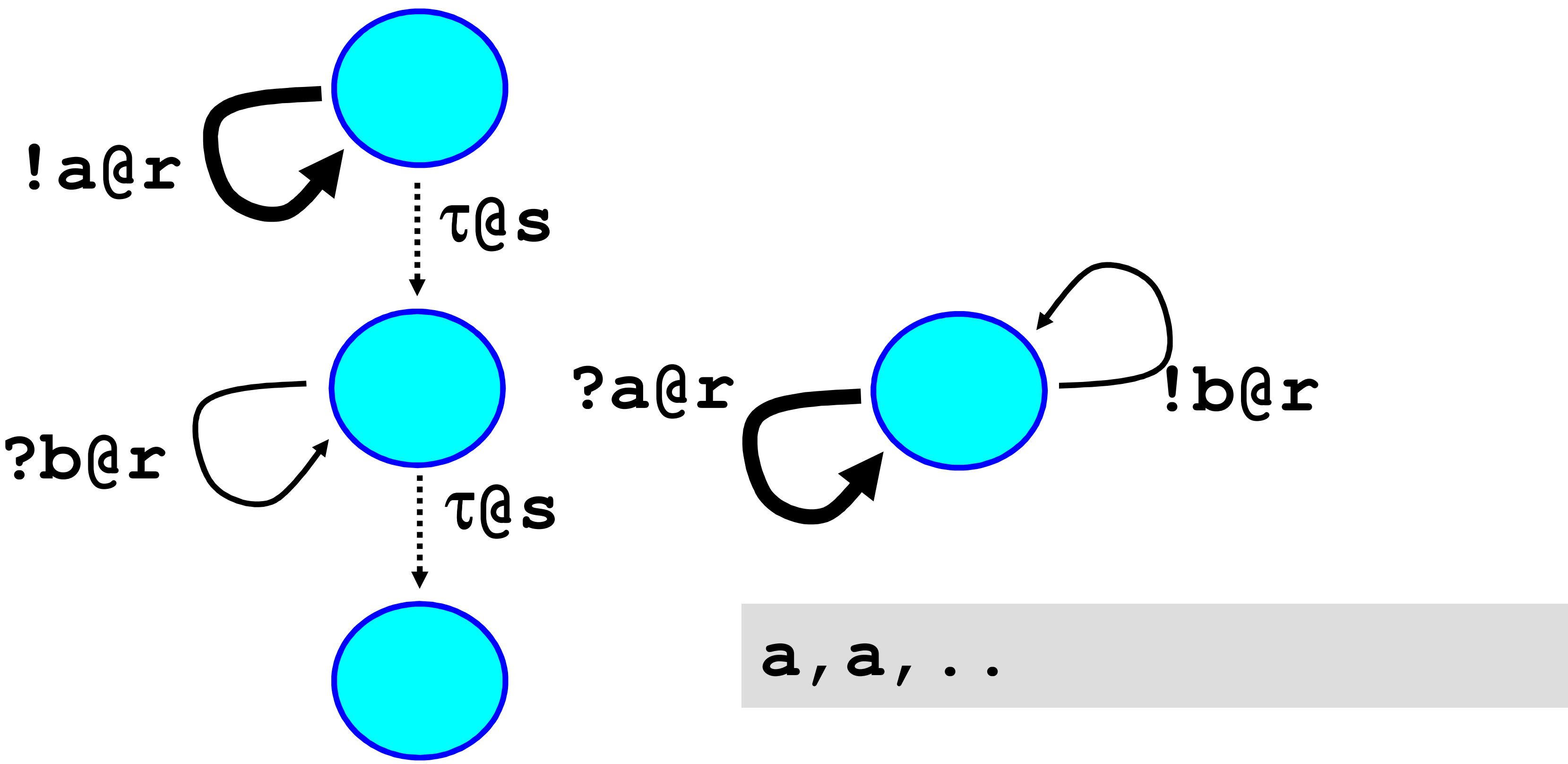
- Starting population: **A|A′**

Example 1: Does it Halt?



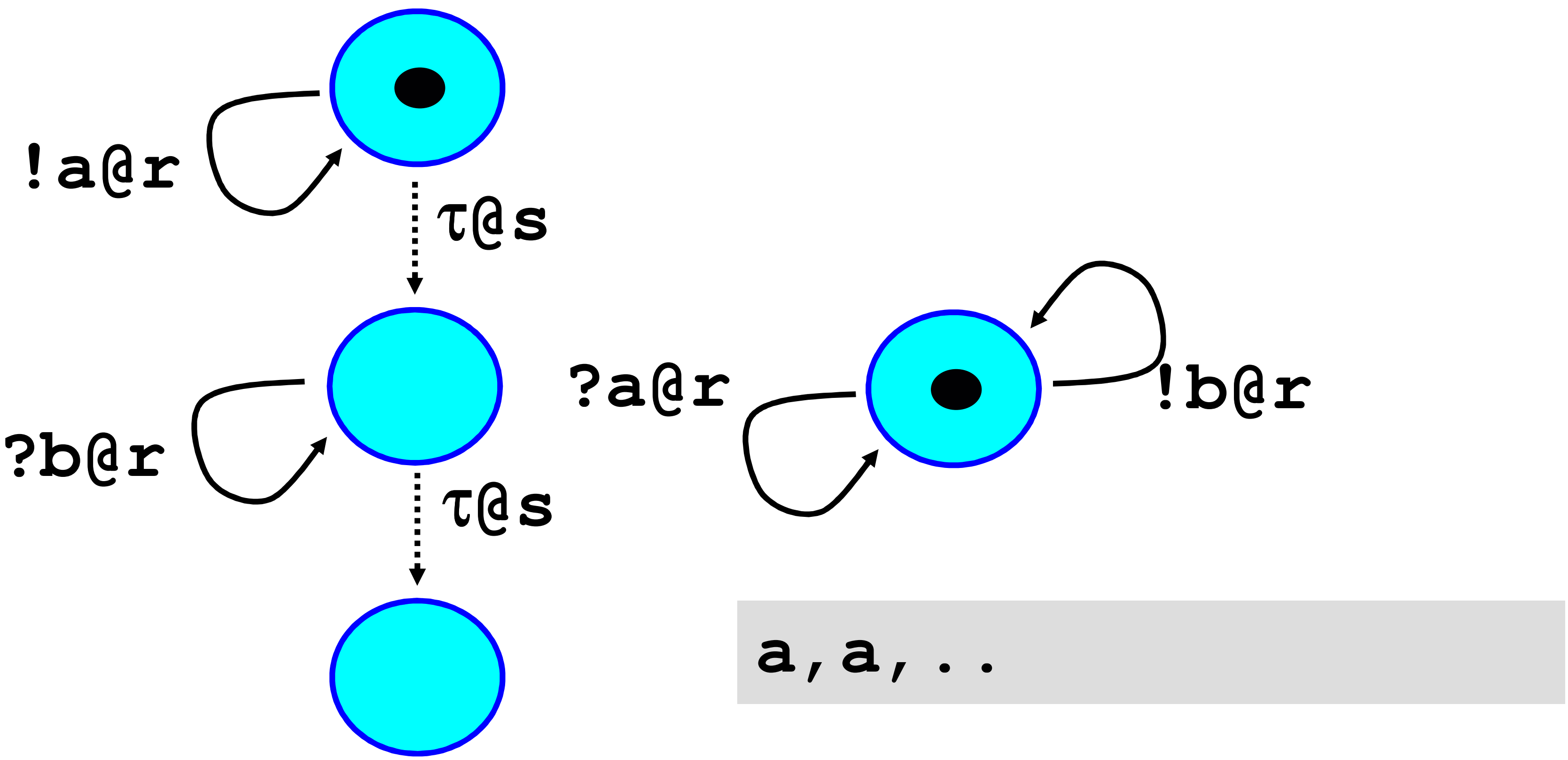
- Starting population: **A|A′**

Example 1: Does it Halt?



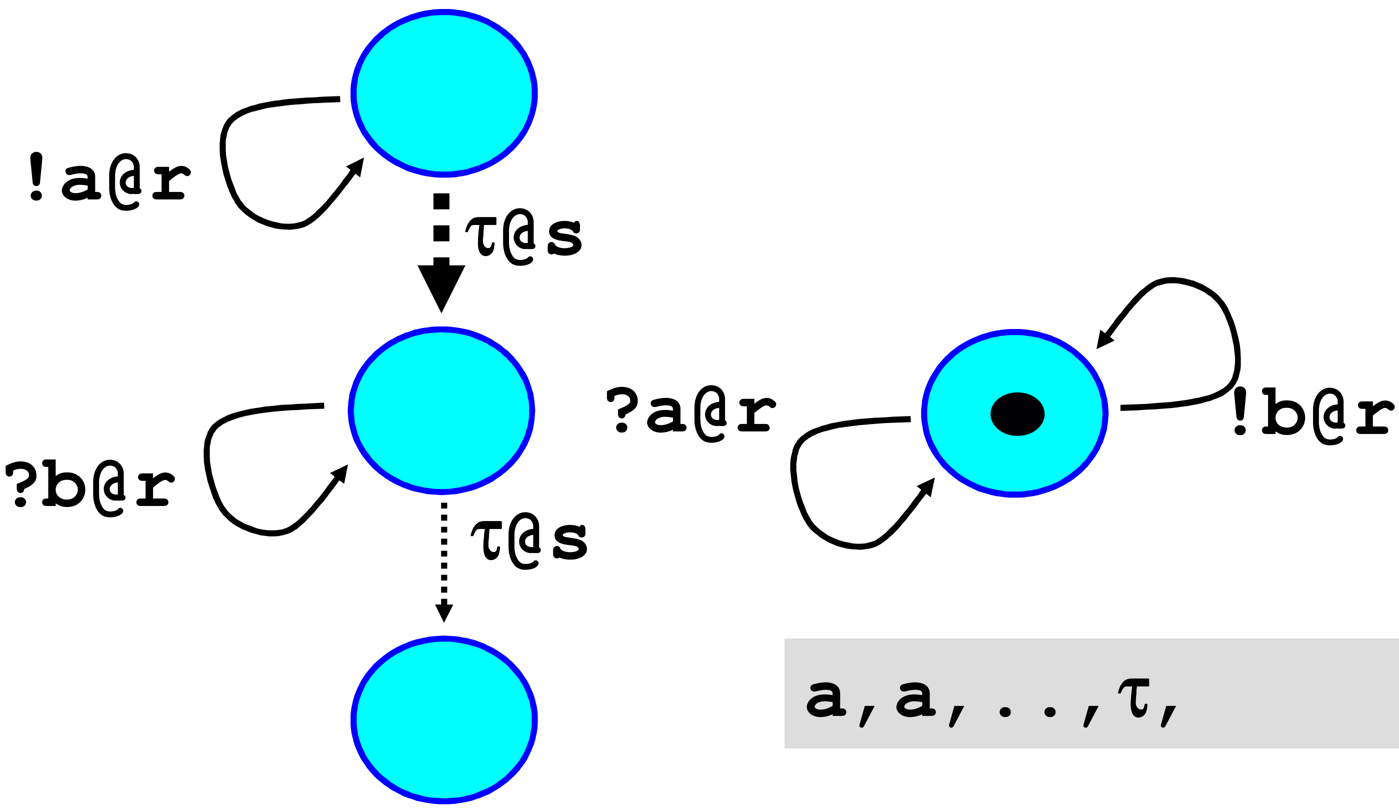
- Starting population: **A|A′**

Example 1: Does it Halt?



- Starting population: **A|A′**

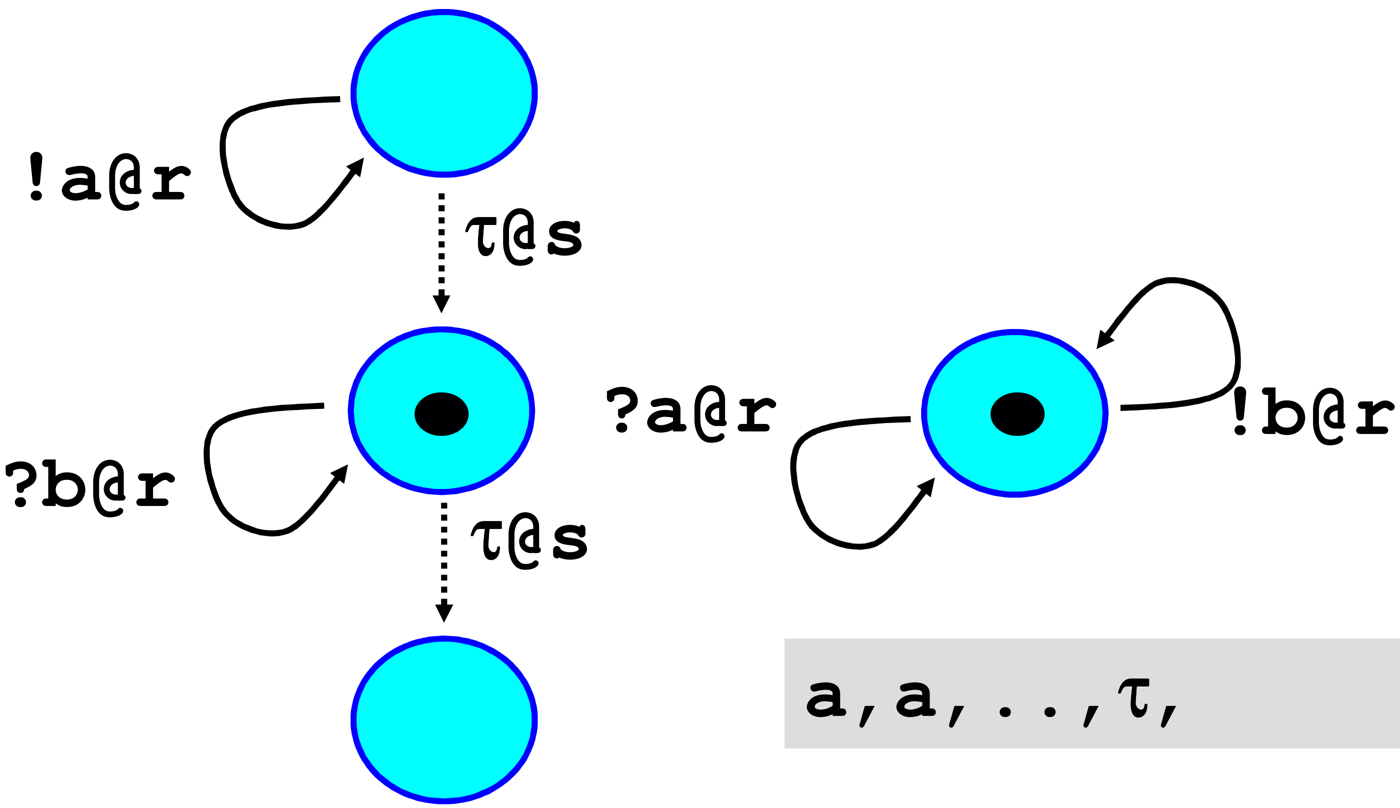
Example 1: Does it Halt?



a, a, . . , τ,

- Starting population: **A|A'**

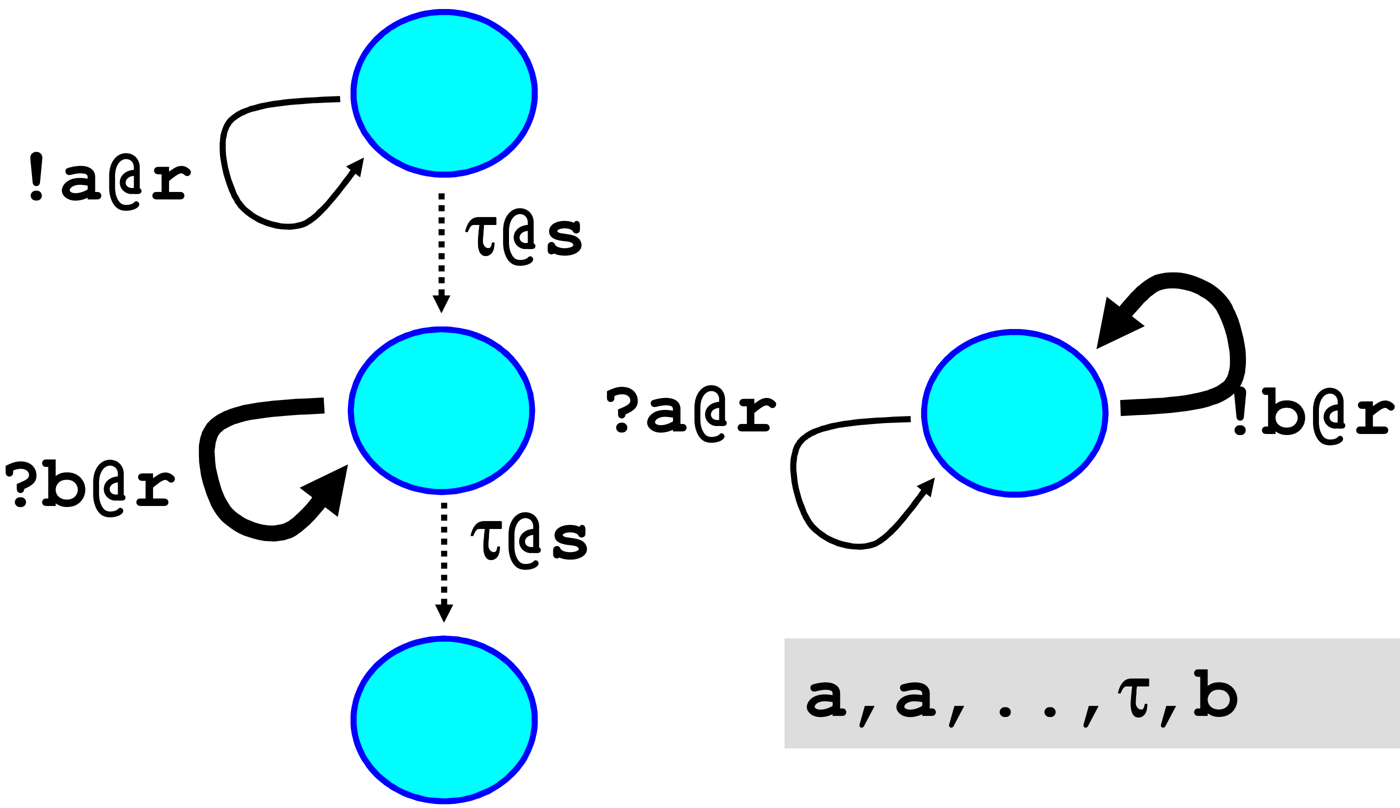
Example 1: Does it Halt?



a,a,..,τ,

- Starting population: **A|A′**

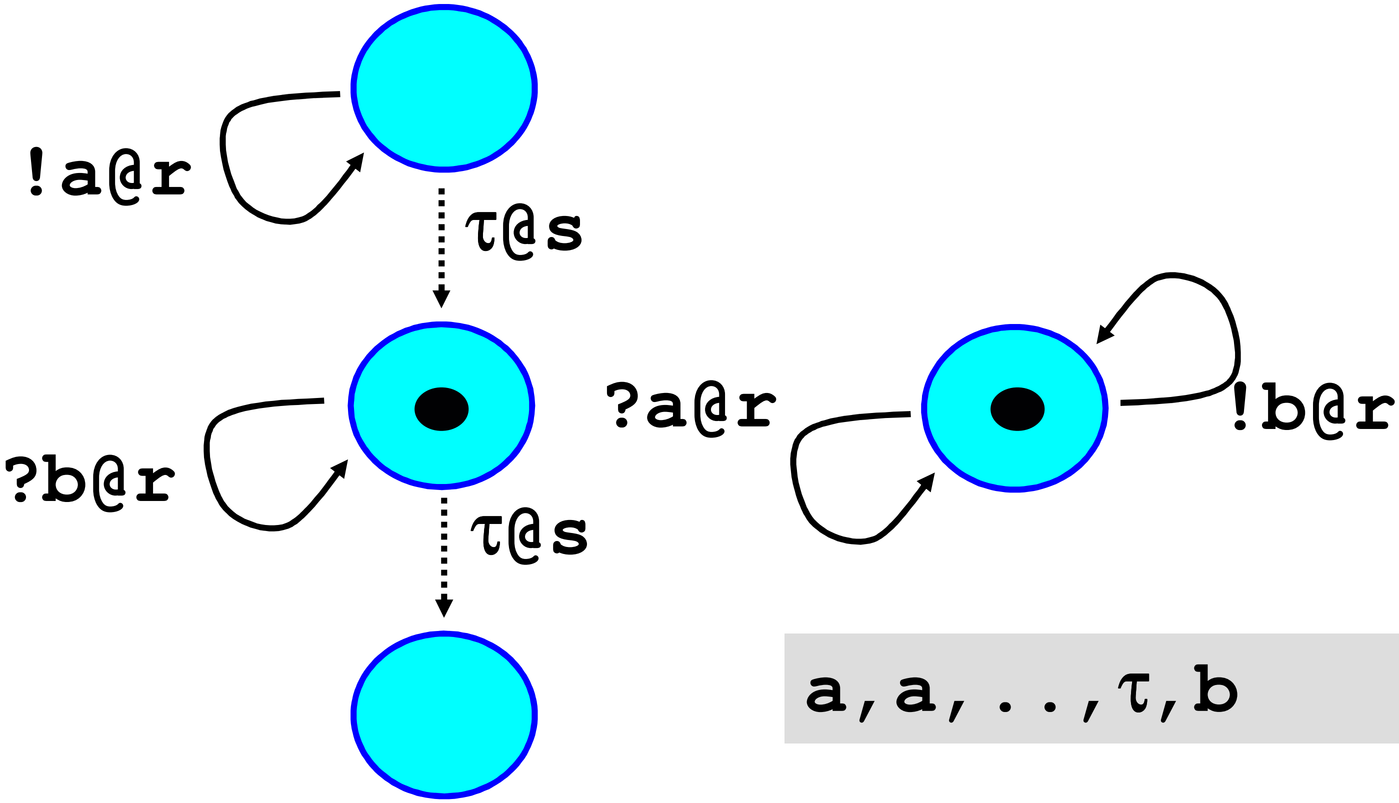
Example 1: Does it Halt?



a,a,..,τ,b

- Starting population: **A|A′**

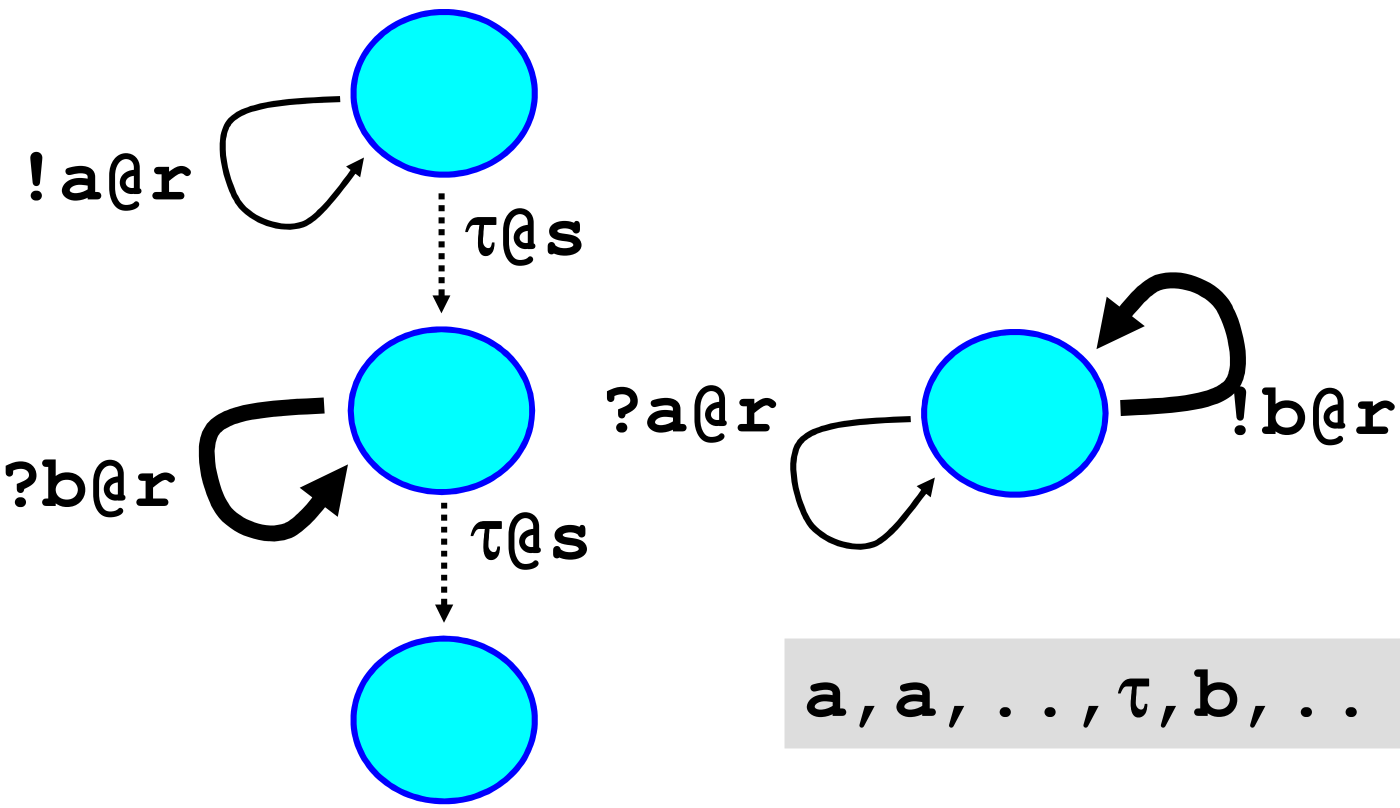
Example 1: Does it Halt?



a,a,..,τ,b

- Starting population: **A|A′**

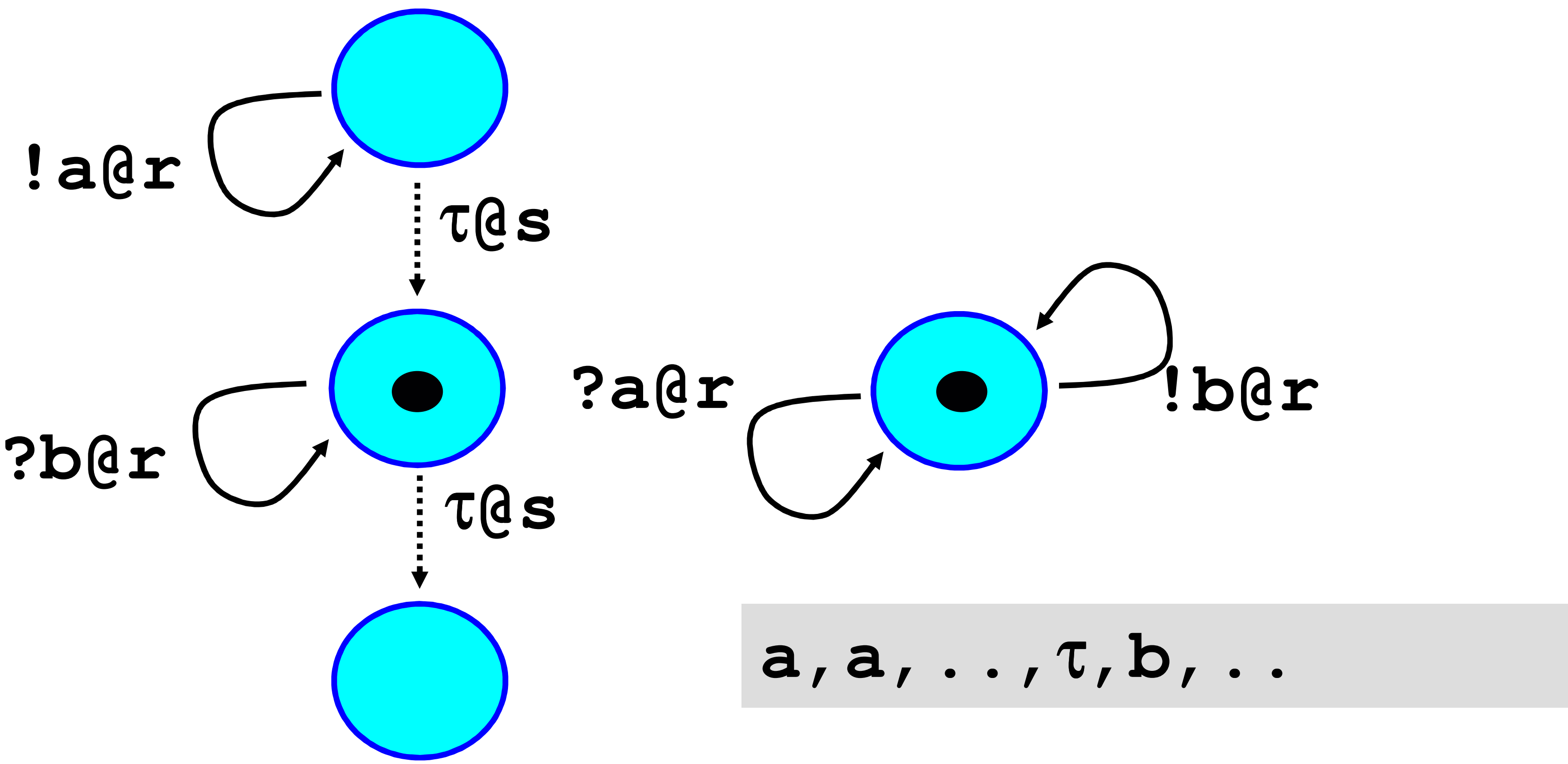
Example 1: Does it Halt?



`a,a,..,τ,b,..`

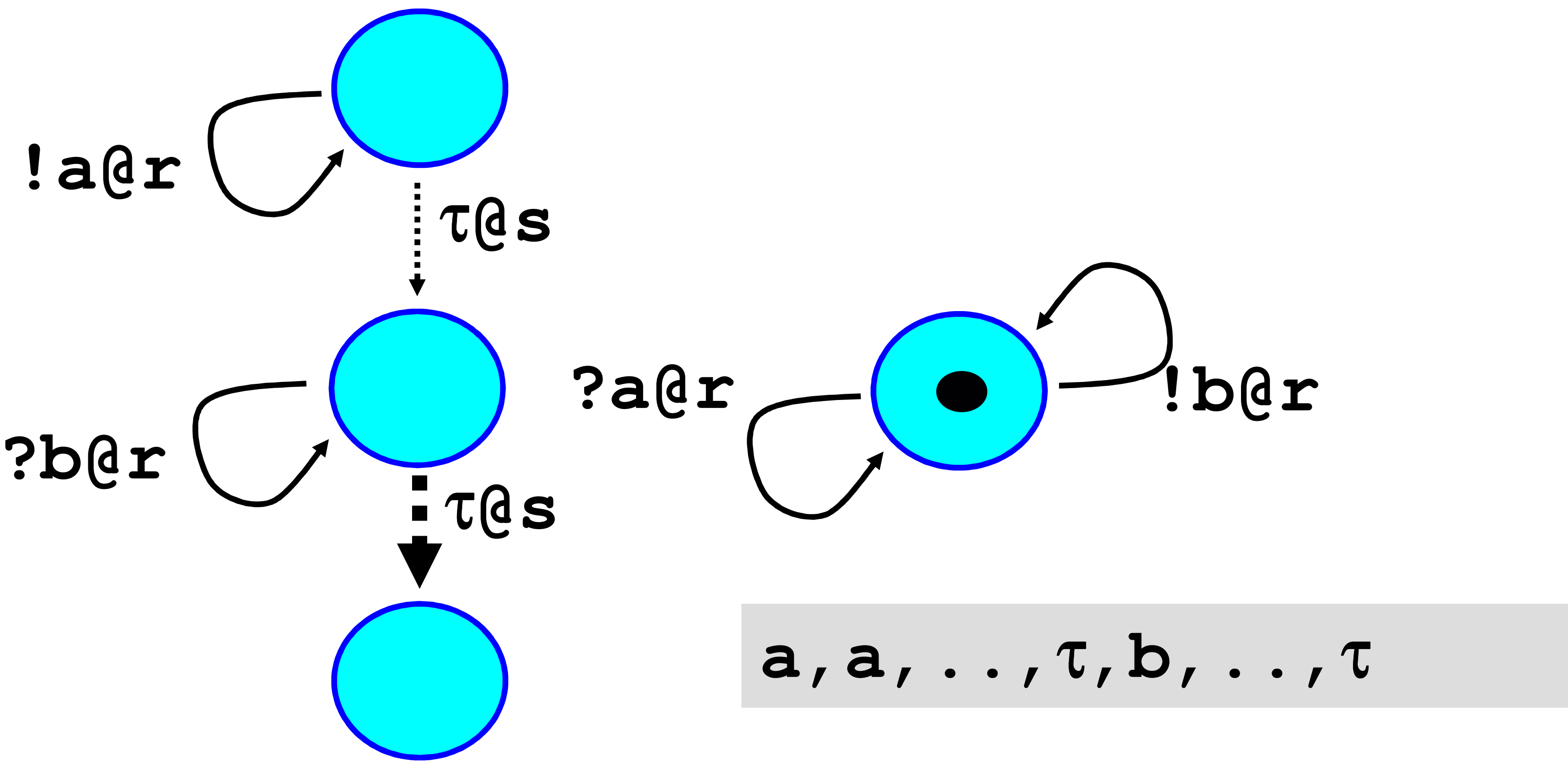
- Starting population: **A|A′**

Example 1: Does it Halt?



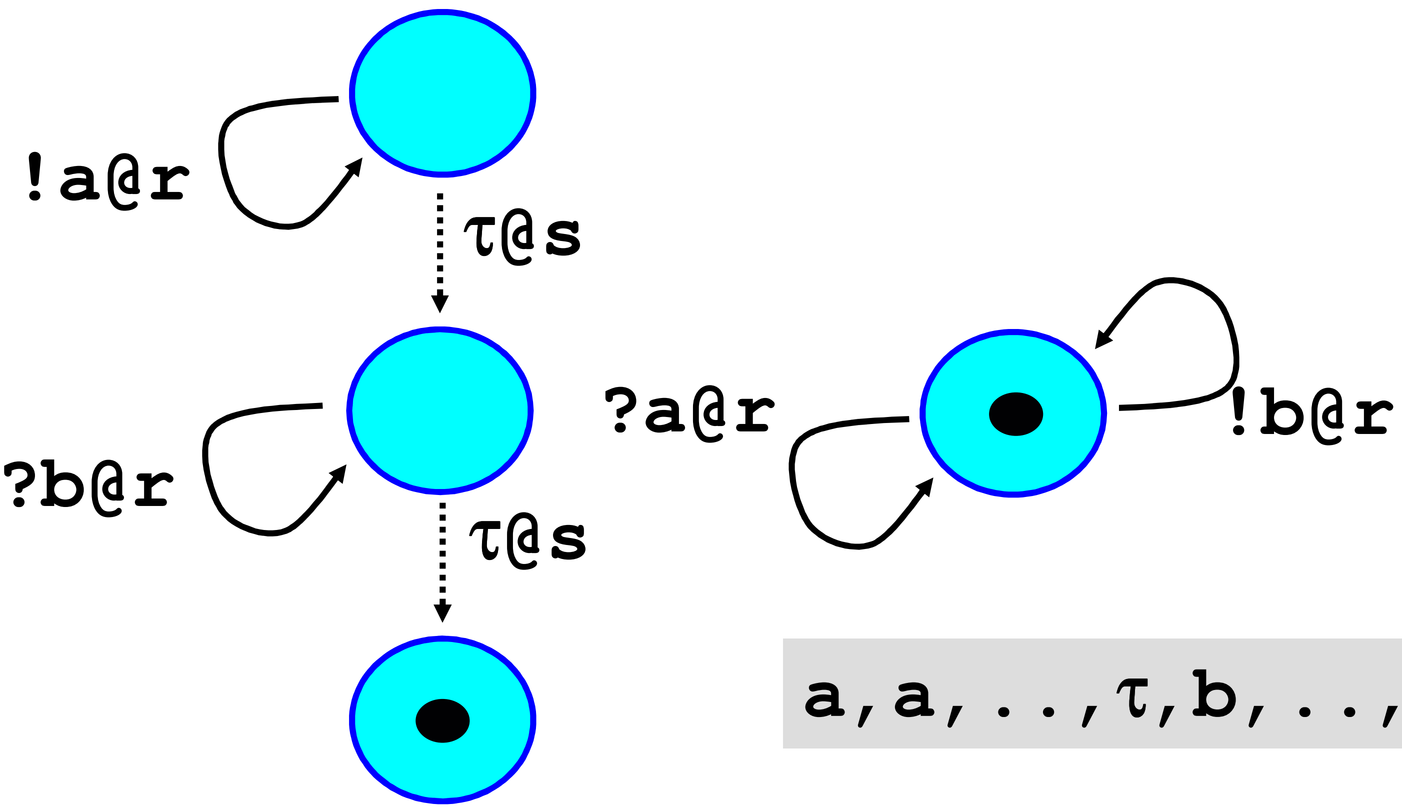
- Starting population: **A|A′**

Example 1: Does it Halt?



- Starting population: **A|A′**

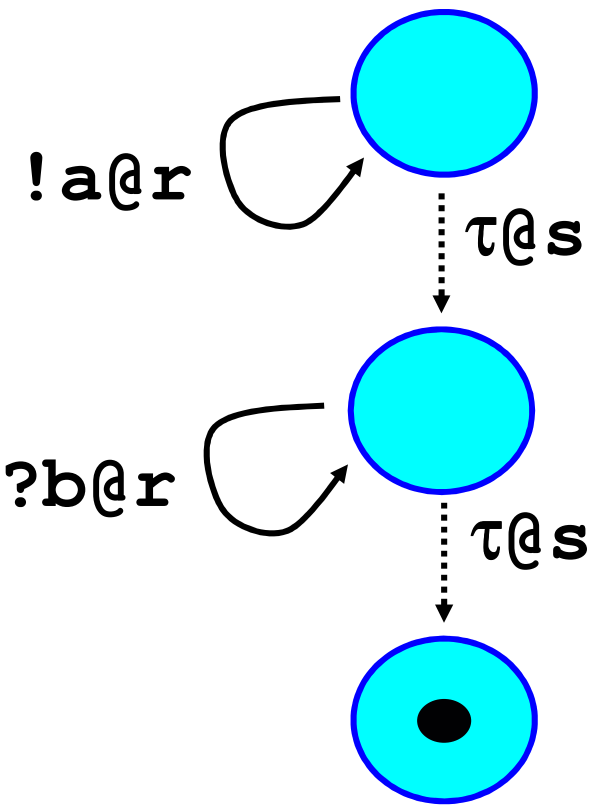
Example 1: Does it Halt?



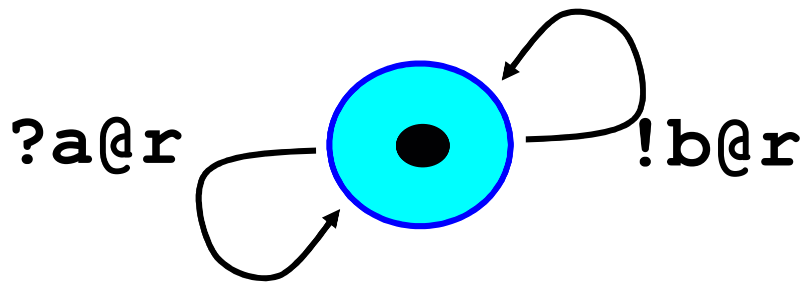
a,a,..,τ,b,..,τ

- Starting population: **A|A′**

Example 1: Does it Halt?



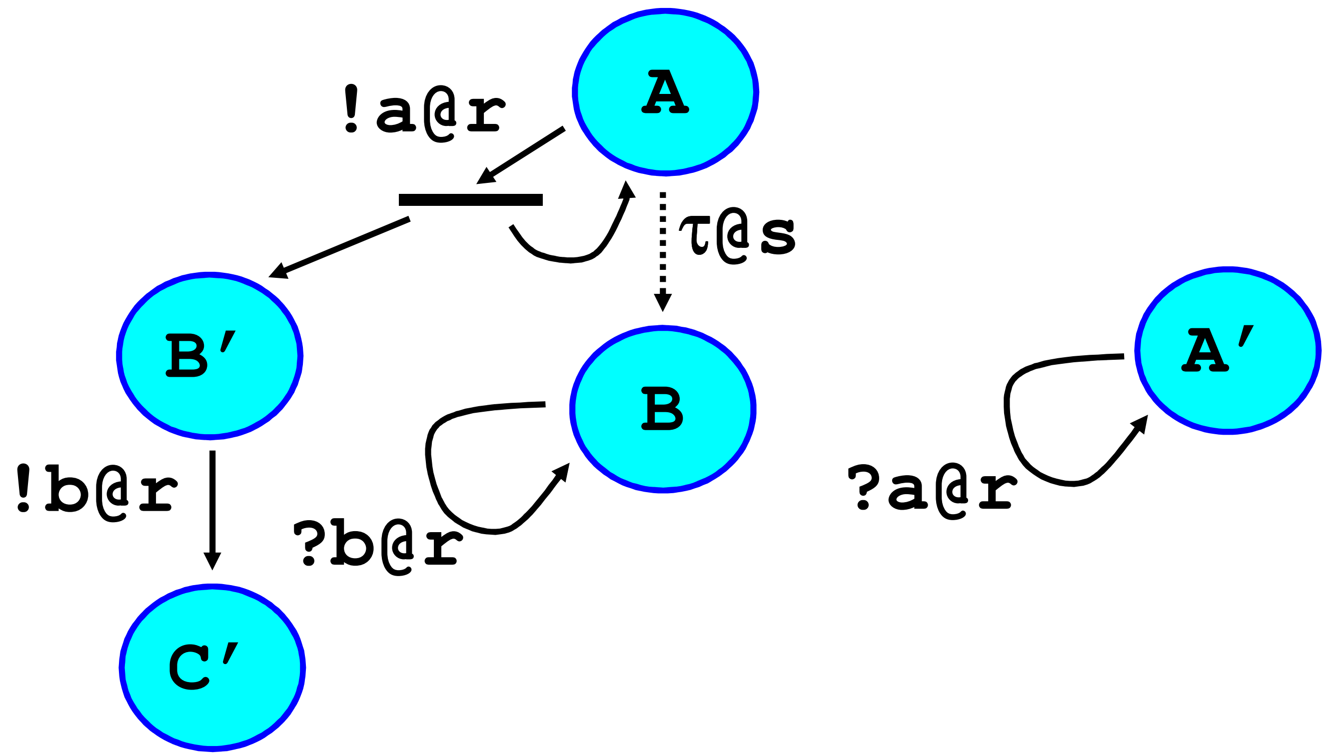
YES



a*τb*τ

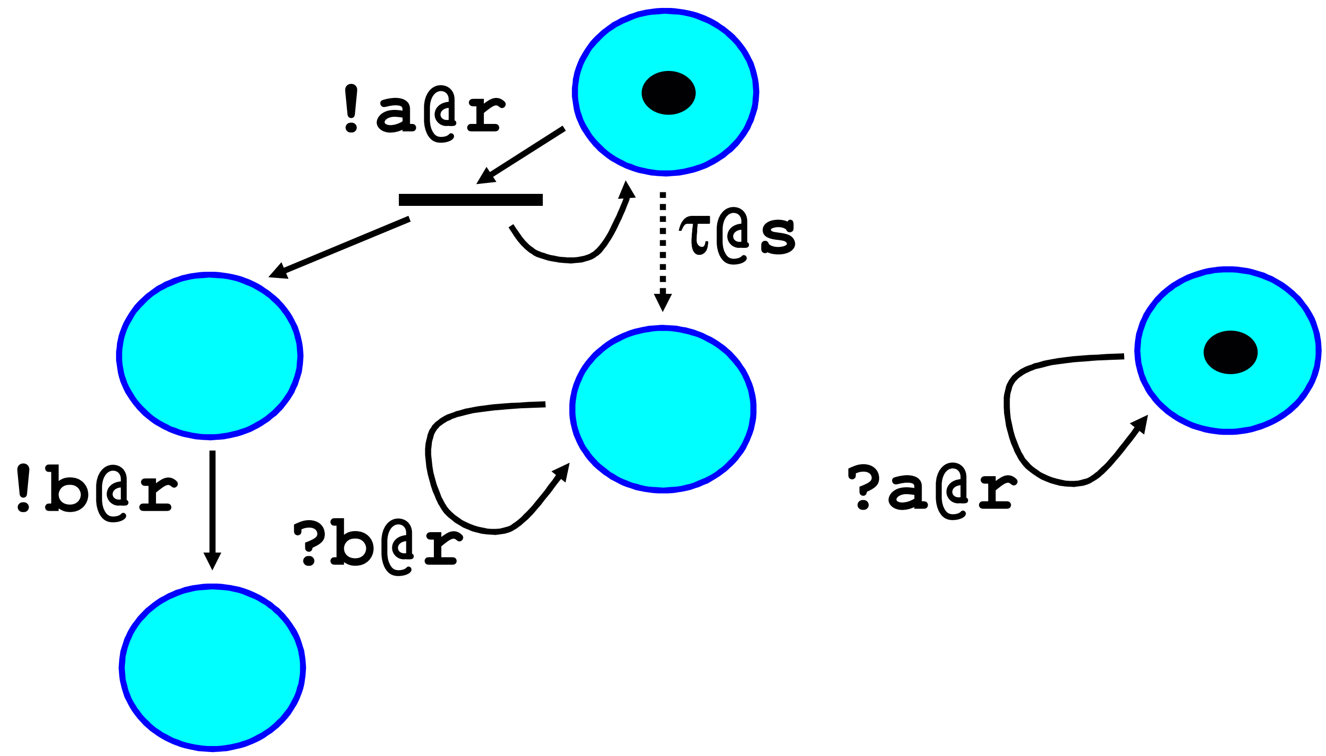
- Starting population: **A|A′**

Example 2: Does it Halt?



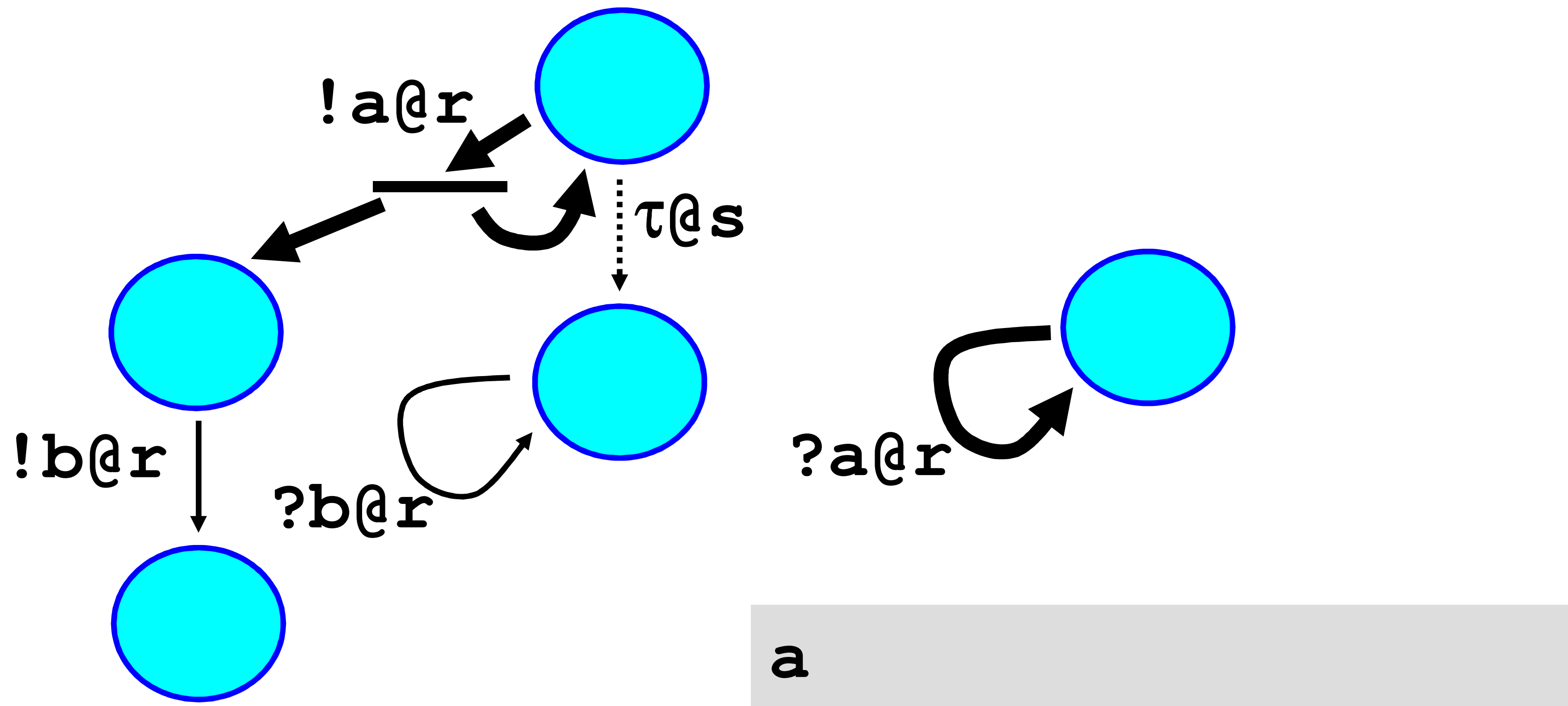
- Starting population: **A|A′**

Example 2: Does it Halt?



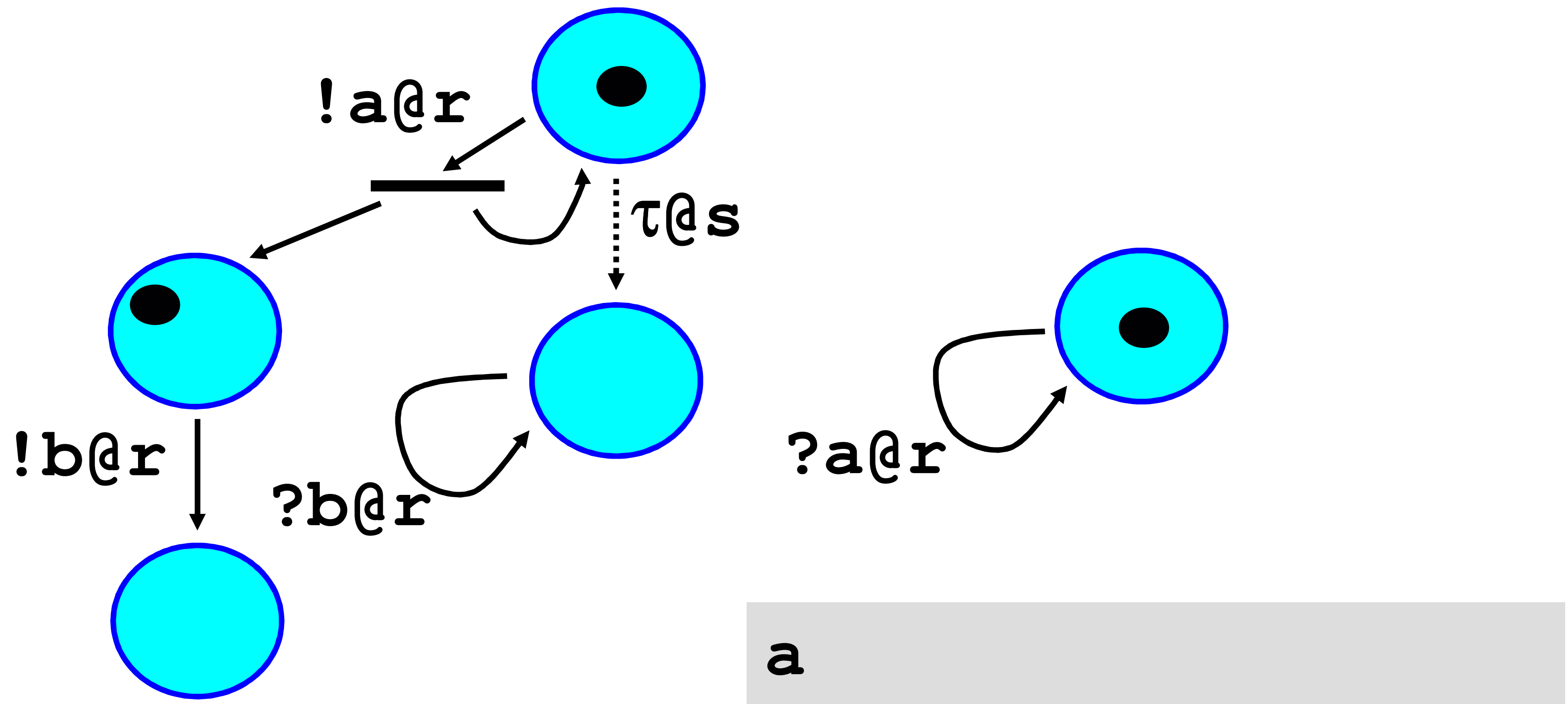
- Starting population: **A|A′**

Example 2: Does it Halt?



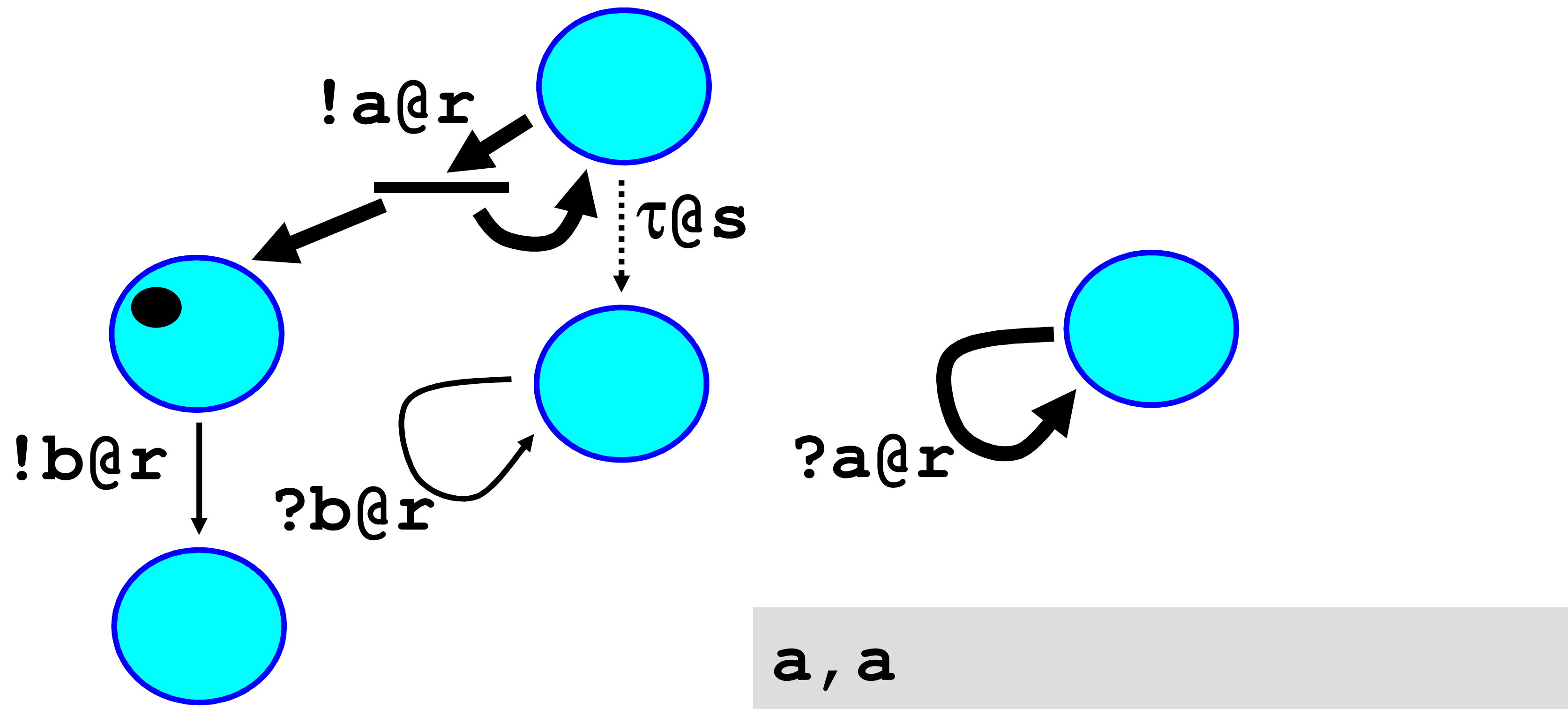
- Starting population: **A|A′**

Example 2: Does it Halt?



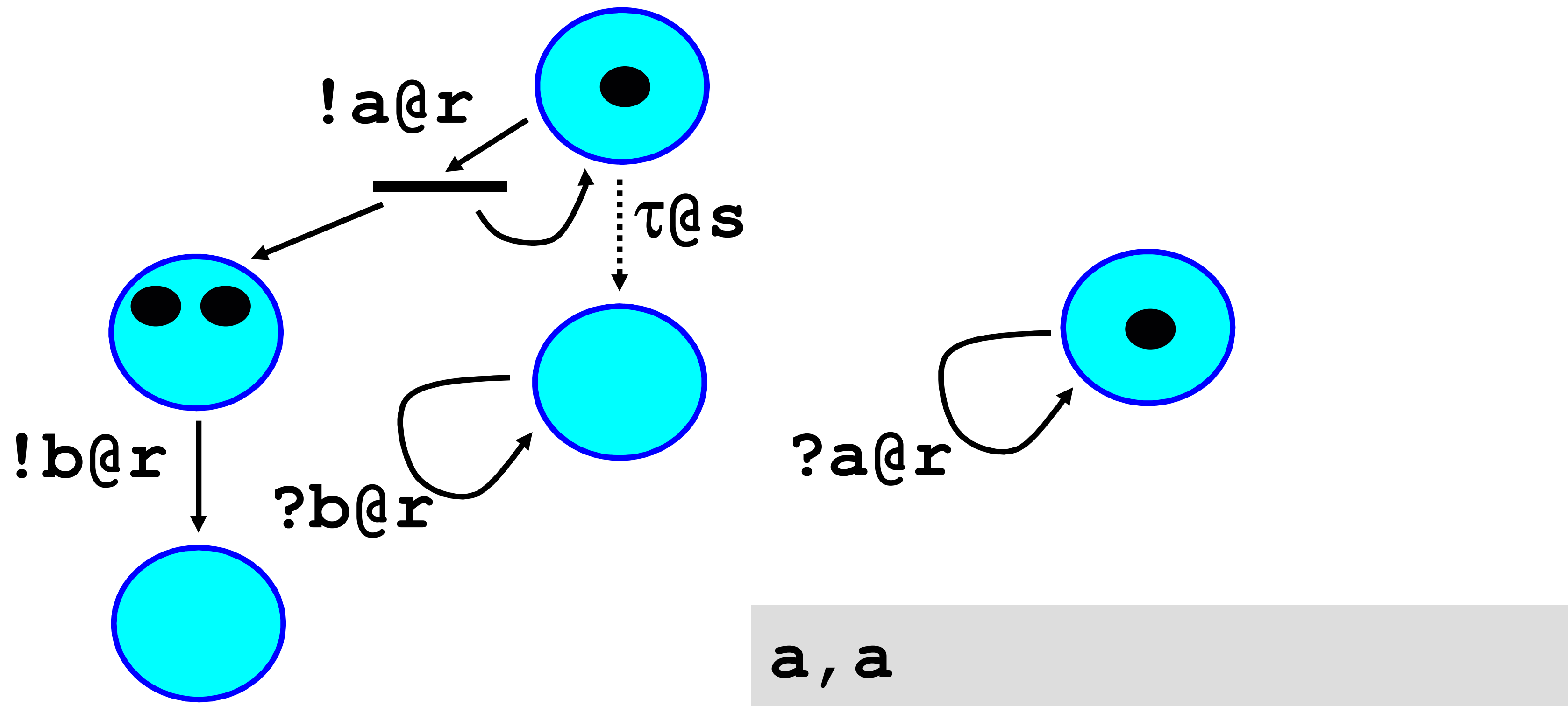
- Starting population: **A|A′**

Example 2: Does it Halt?



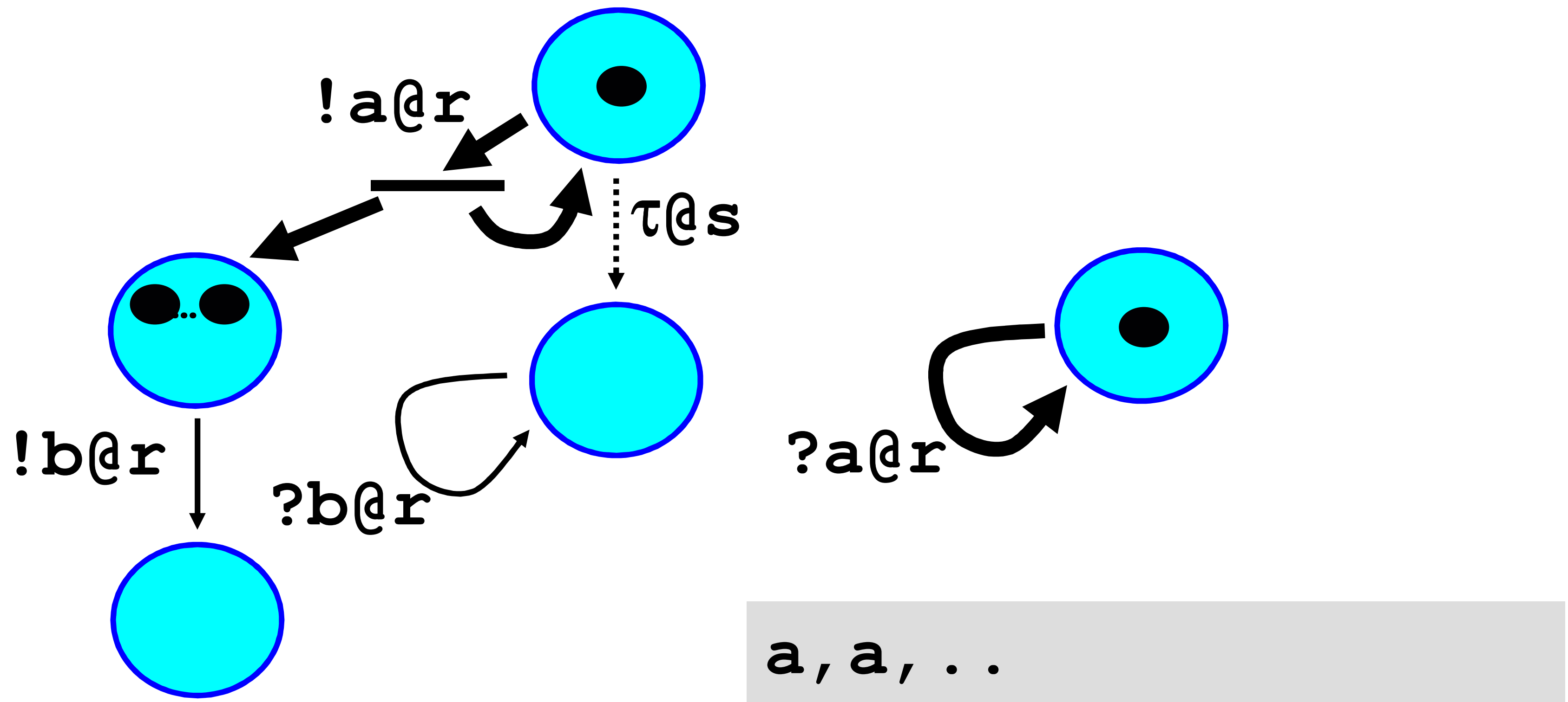
- Starting population: **A|A′**

Example 2: Does it Halt?



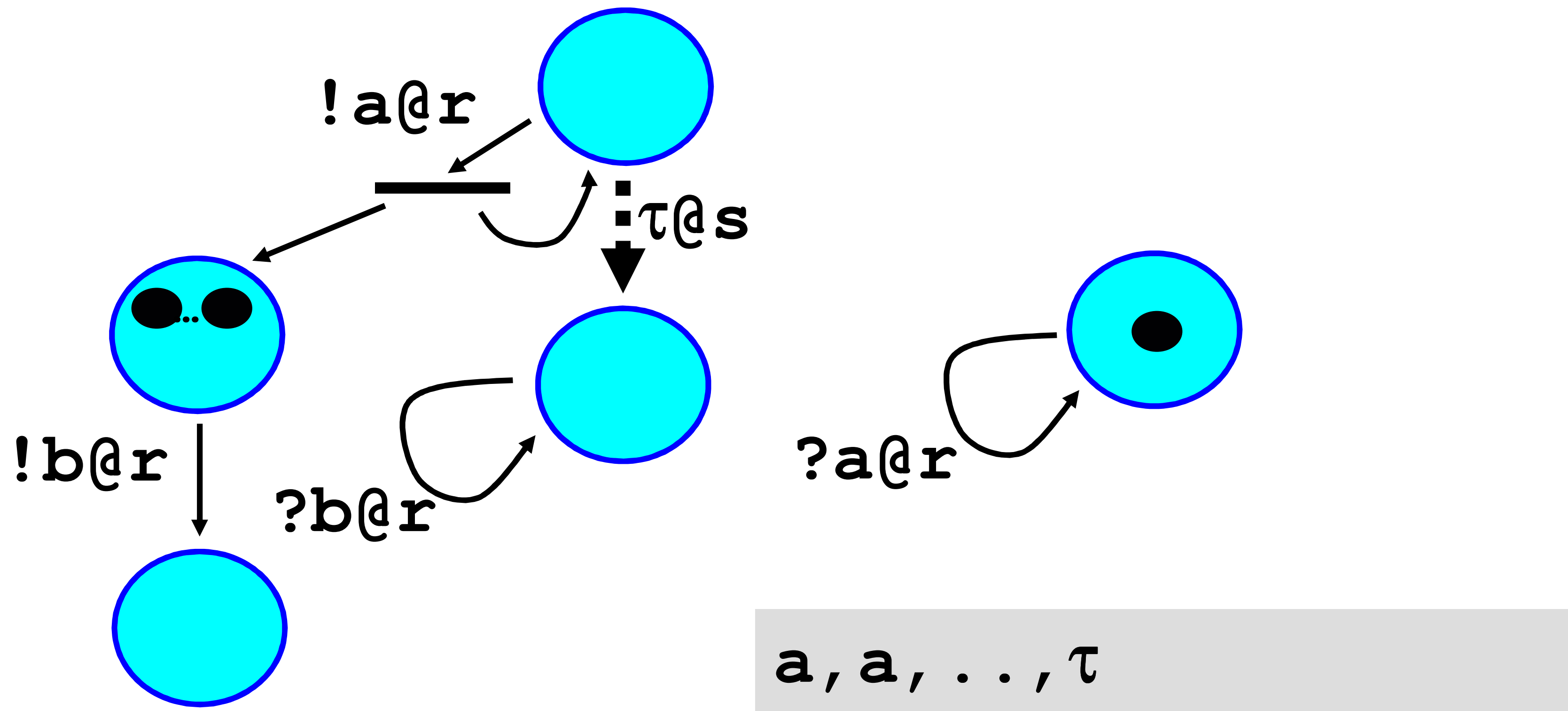
- Starting population: **A|A′**

Example 2: Does it Halt?



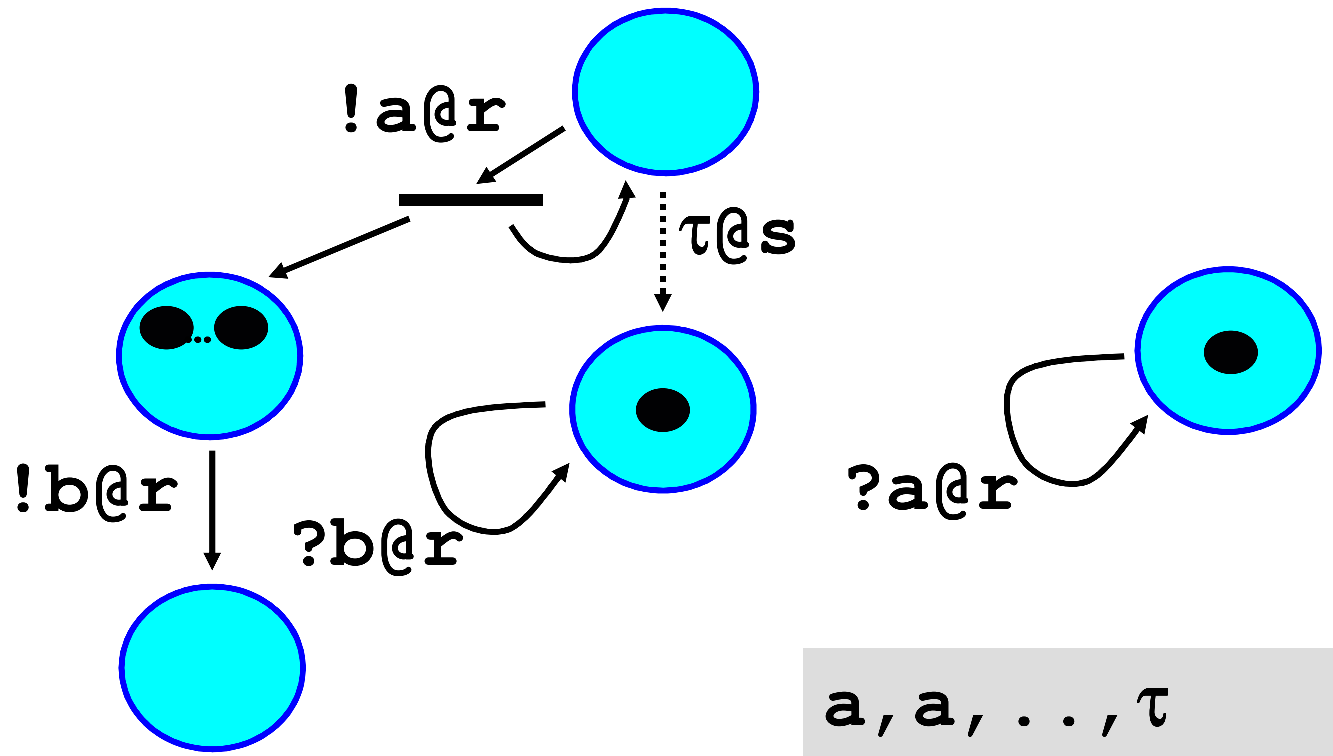
- Starting population: **A|A′**

Example 2: Does it Halt?



- Starting population: **A|A′**

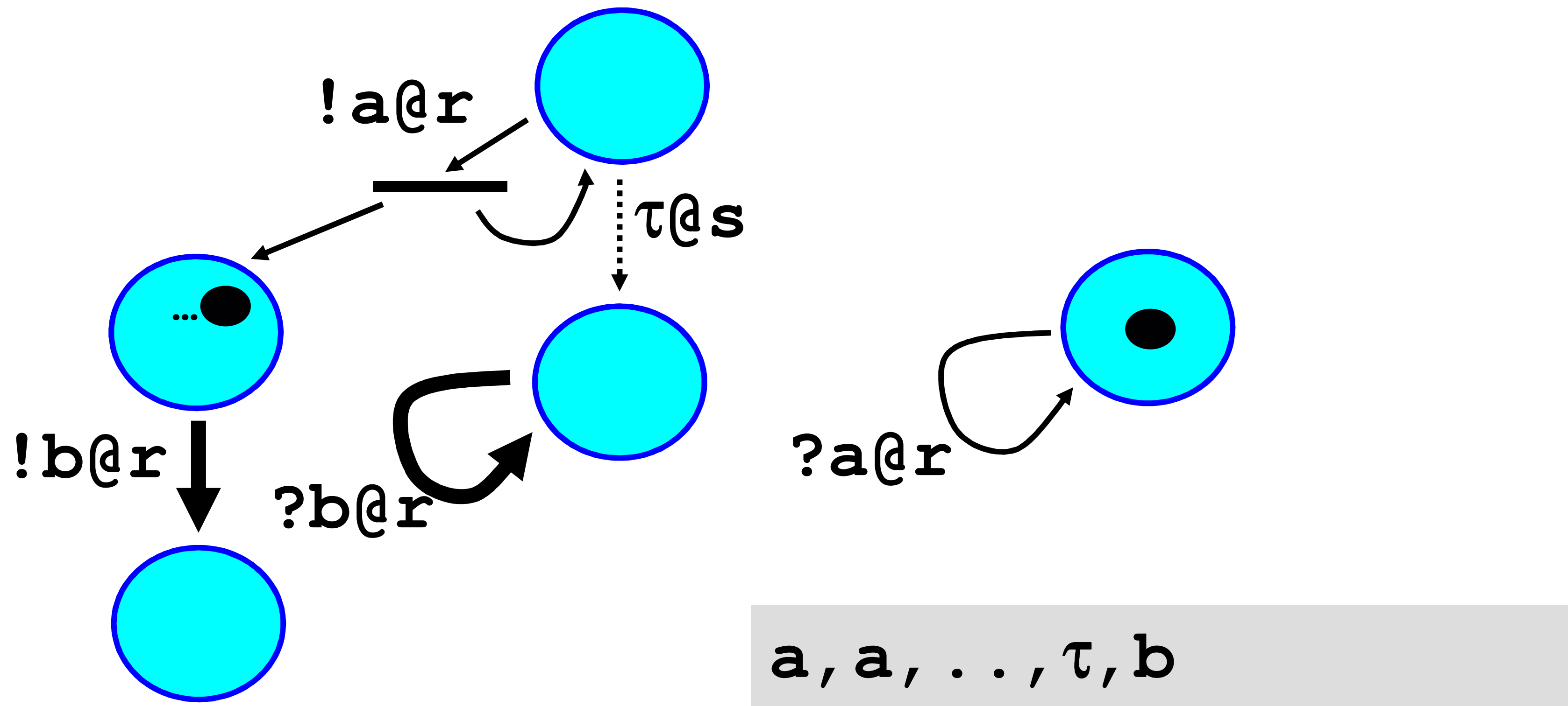
Example 2: Does it Halt?



a,a, . . ,τ

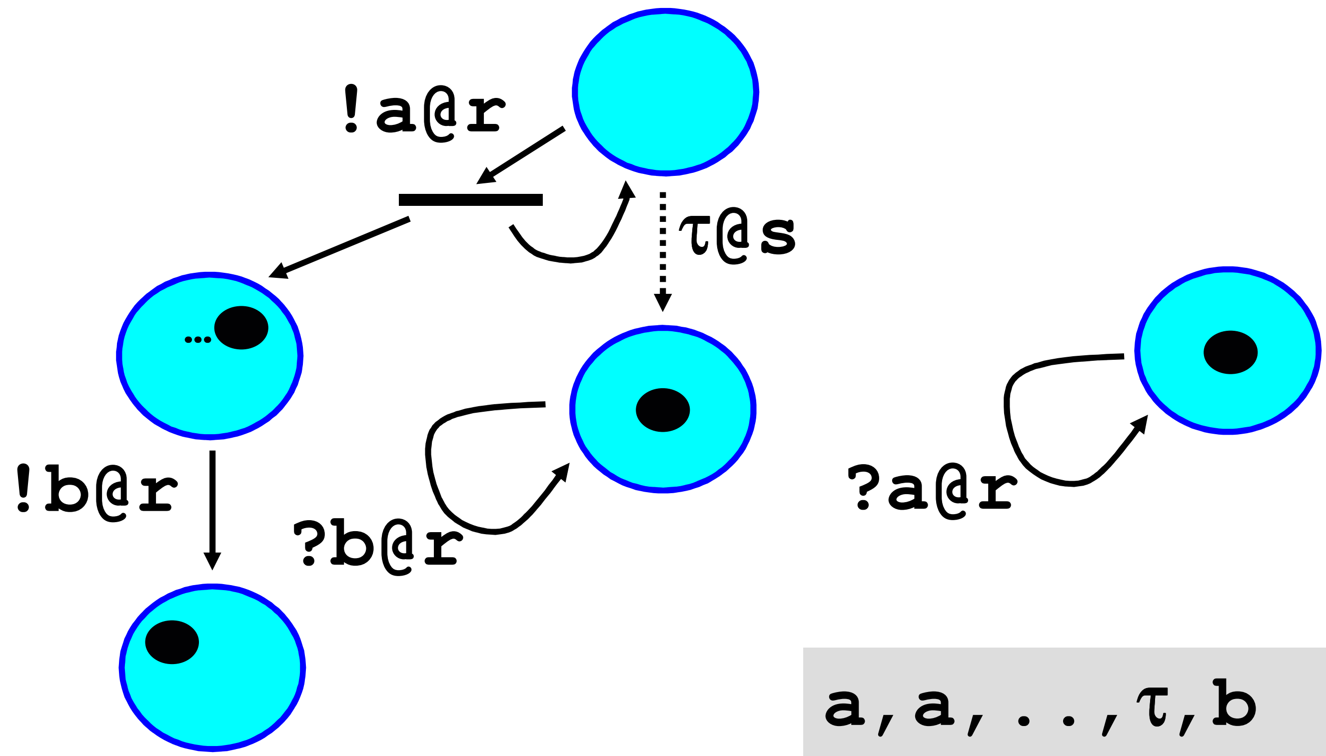
- Starting population: **A|A′**

Example 2: Does it Halt?



- Starting population: **A|A′**

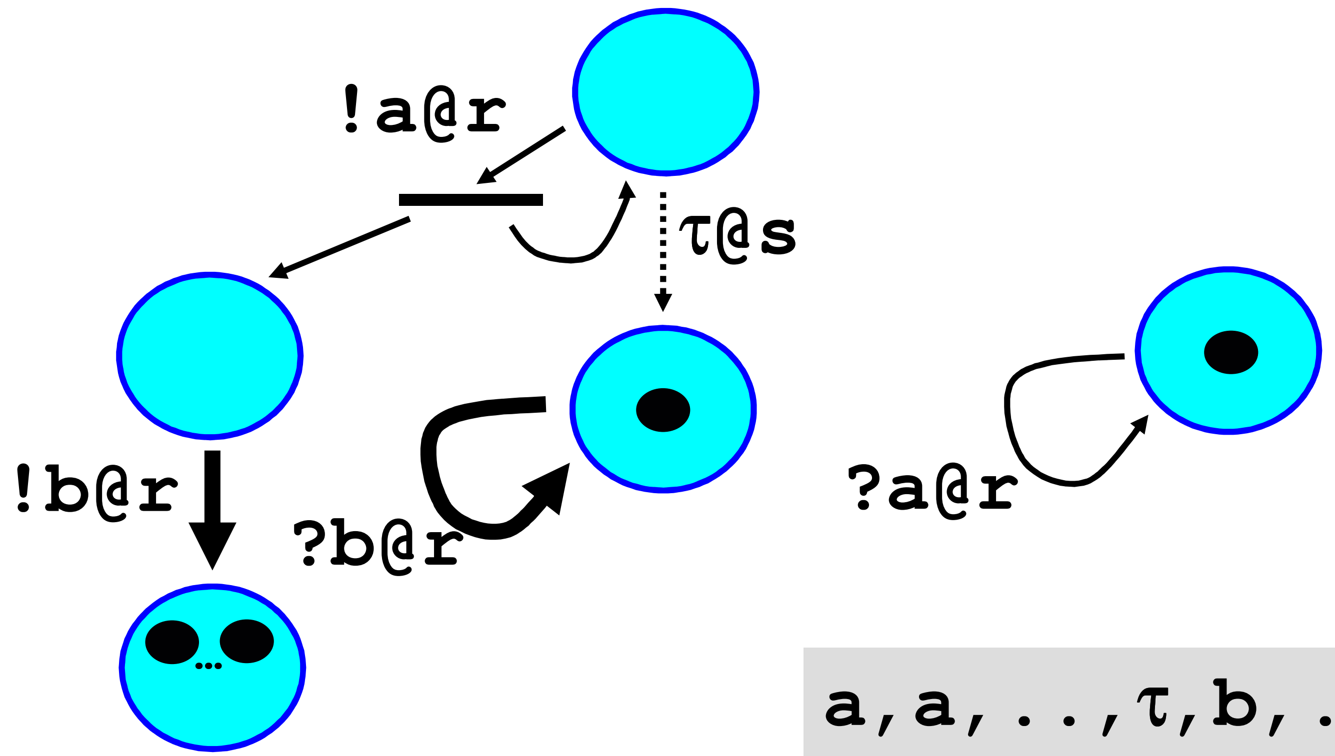
Example 2: Does it Halt?



a,a,..,τ,b

- Starting population: **A|A′**

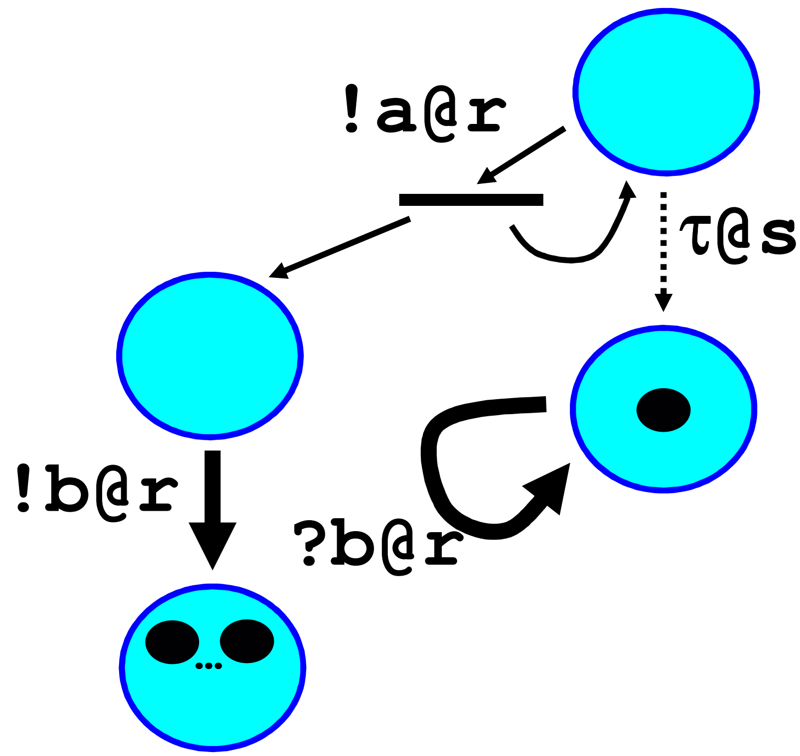
Example 2: Does it Halt?



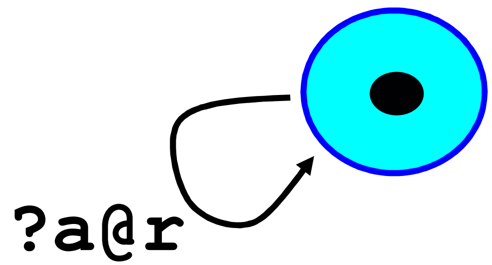
a,a,..,τ,b,..,b

- Starting population: **A|A′**

Example 2: Does it Halt?



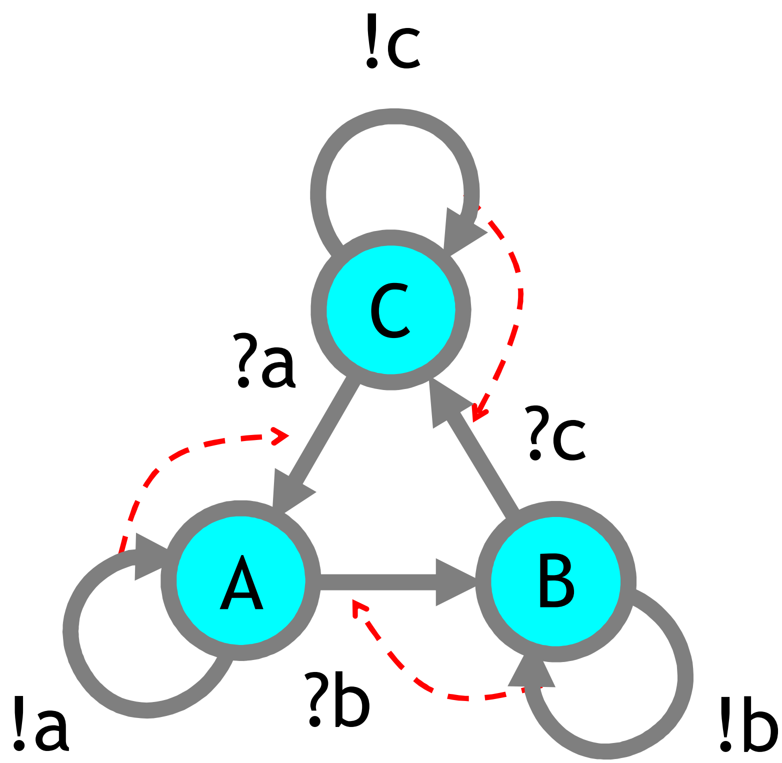
YES



$a^n \tau b^n$

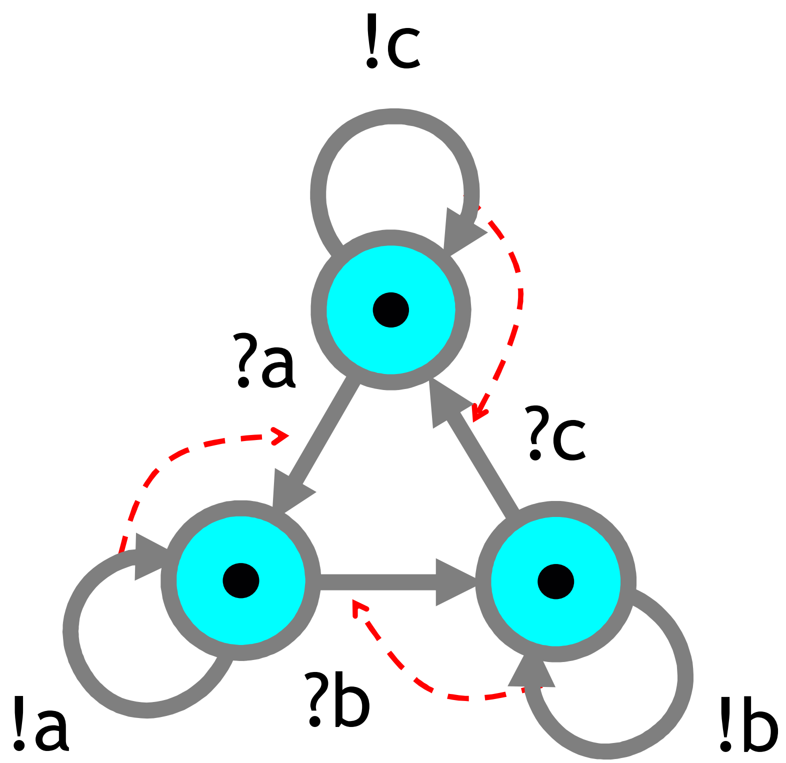
- Starting population: **A|A′**

Example 3. Does it halt?



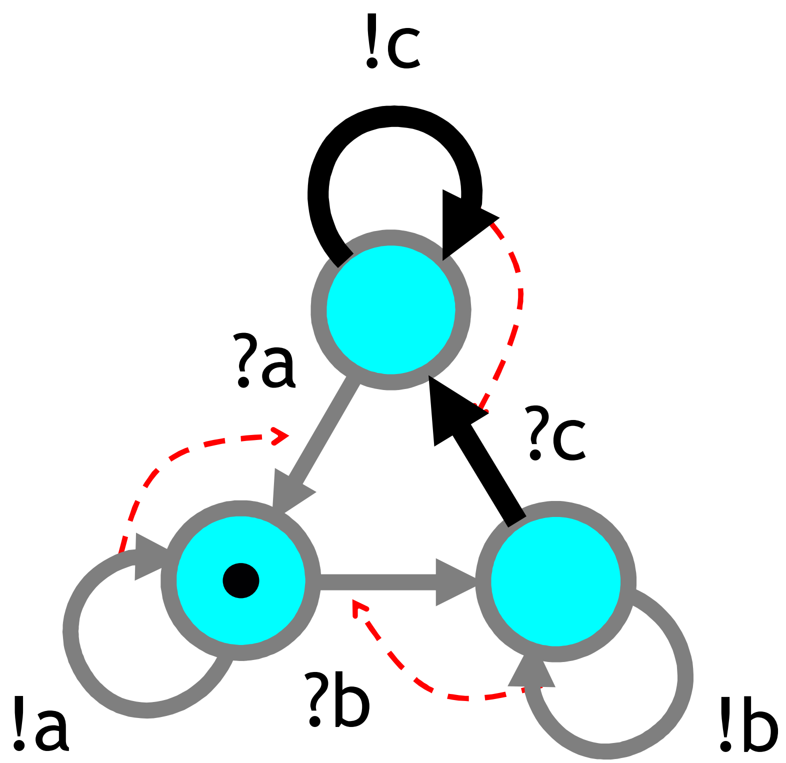
Example 3. Does it halt?

3 Automata



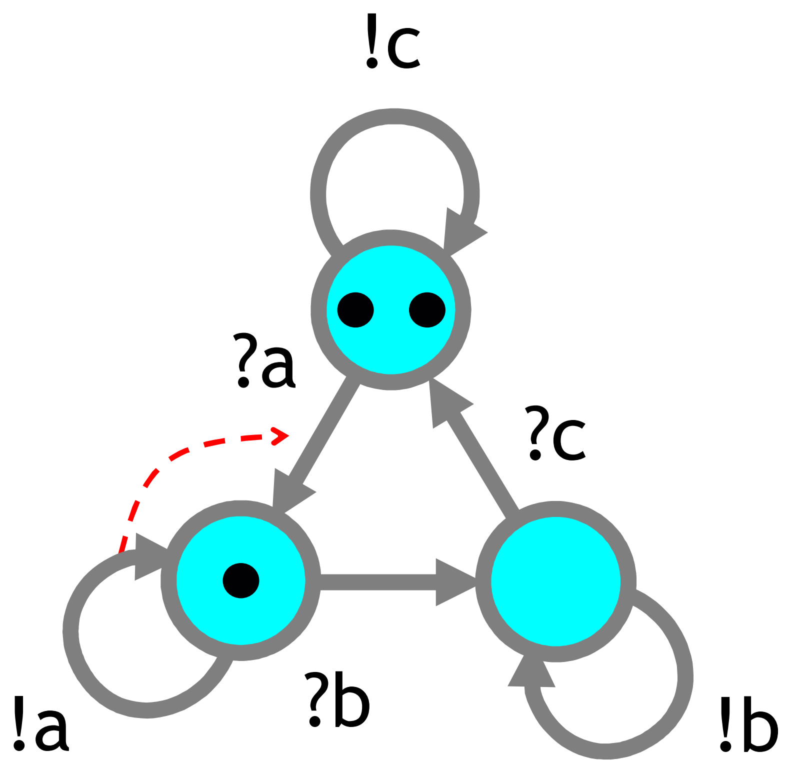
Example 3. Does it halt?

3 Automata



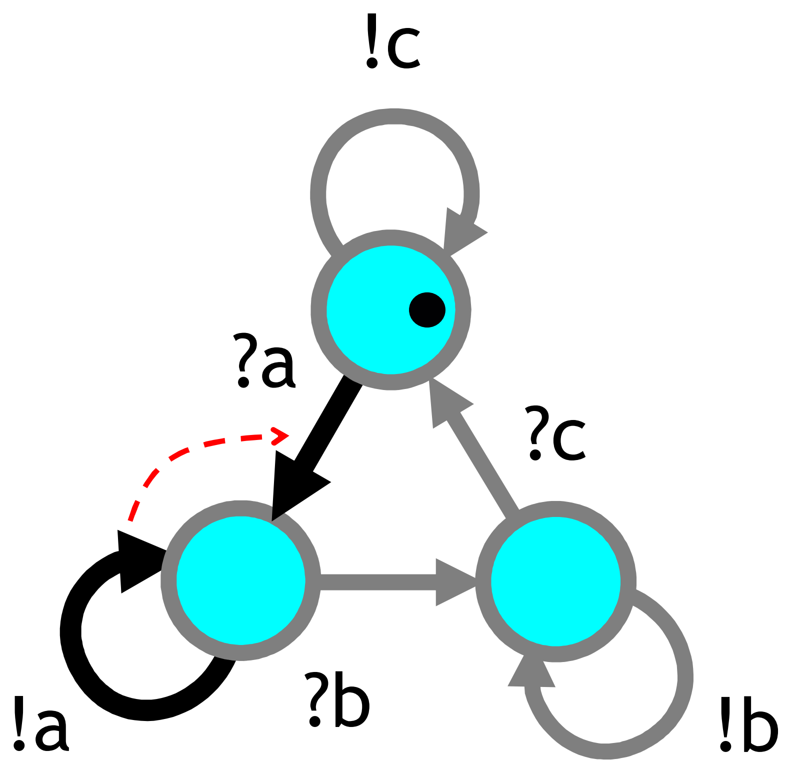
Example 3. Does it halt?

3 Automata



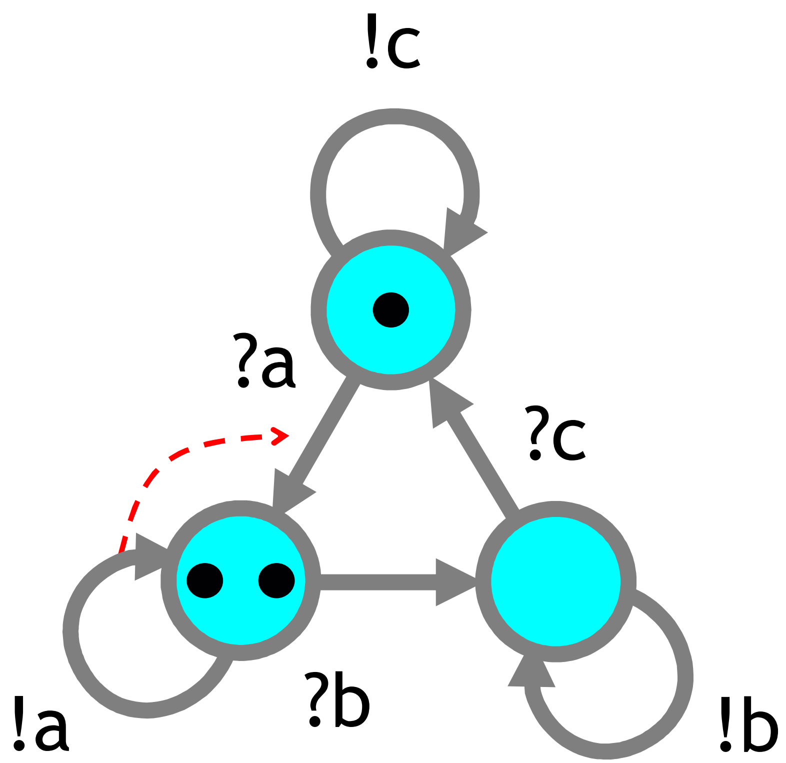
Example 3. Does it halt?

3 Automata



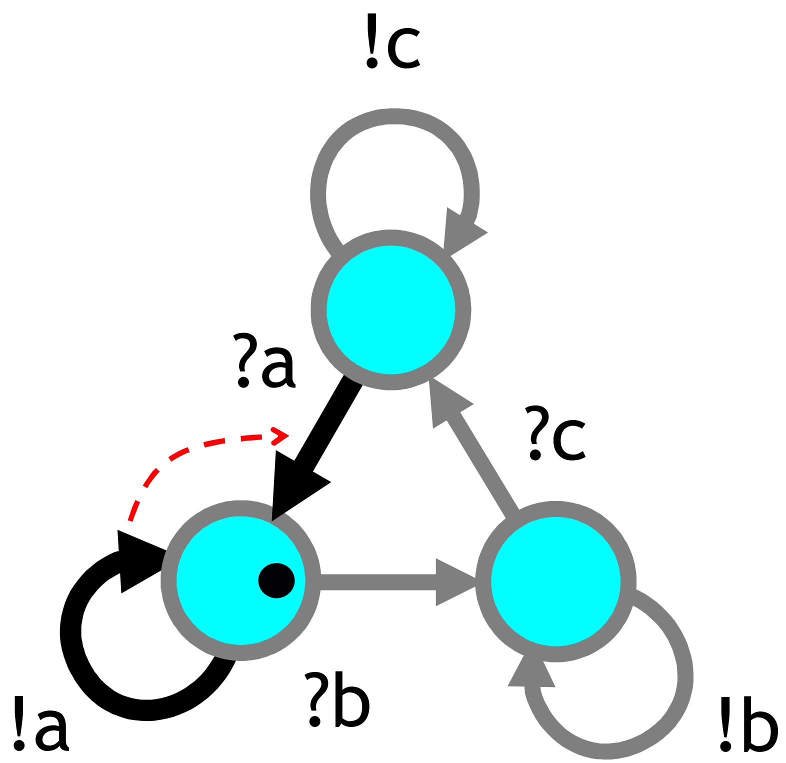
Example 3. Does it halt?

3 Automata



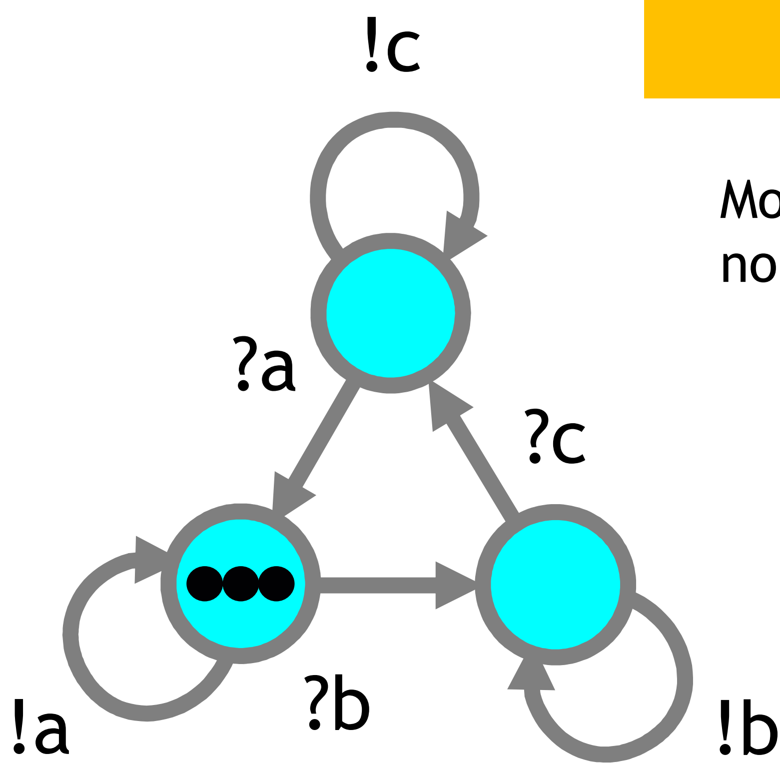
Example 3. Does it halt?

3 Automata



Example 3. Does it halt?

3 Automata

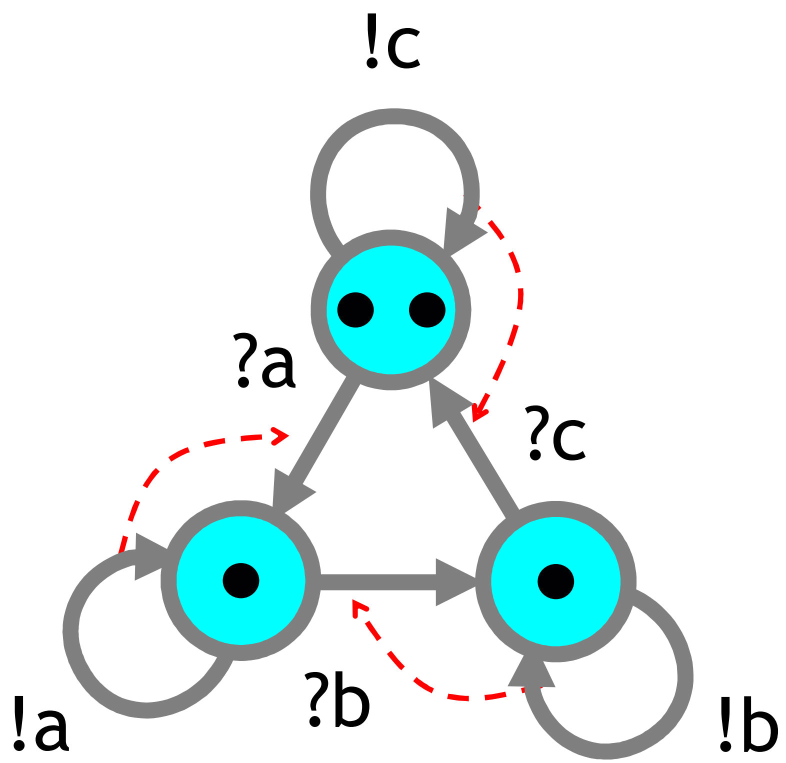


YES

Moreover, there is no infinite trace.

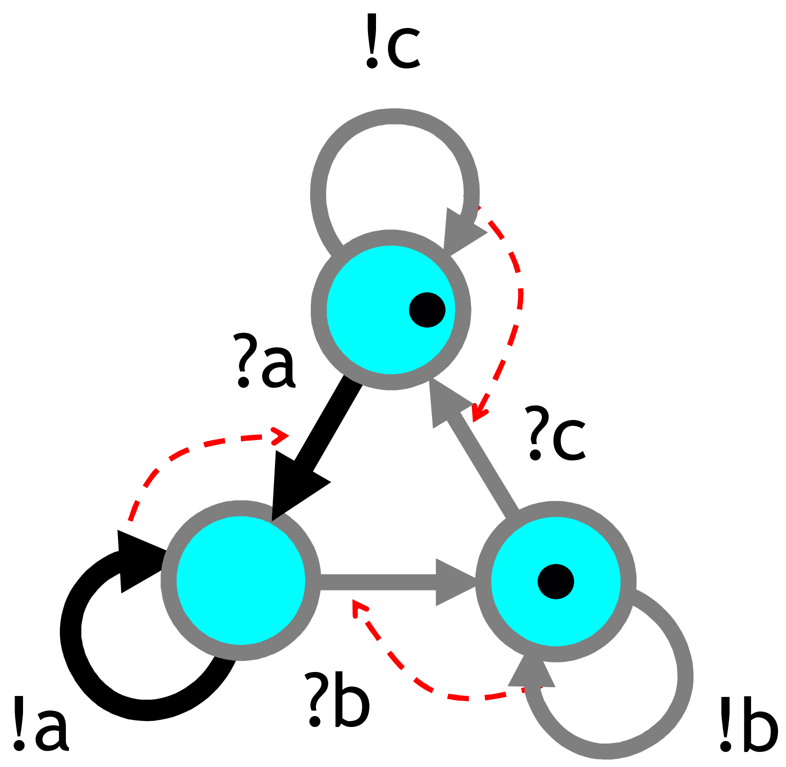
Example 3. Does it halt?

4 Automata



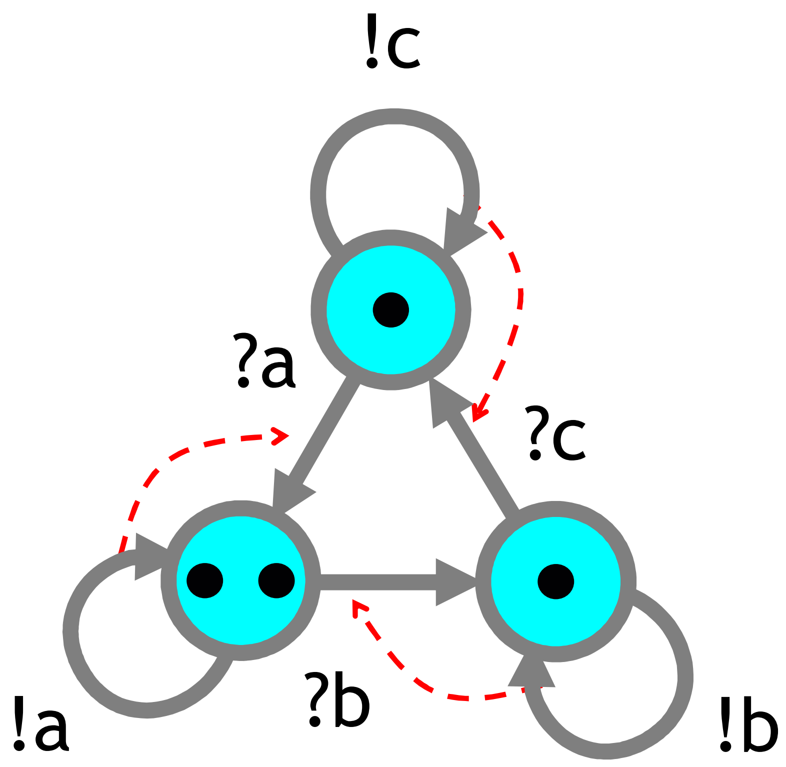
Example 3. Does it halt?

4 Automata



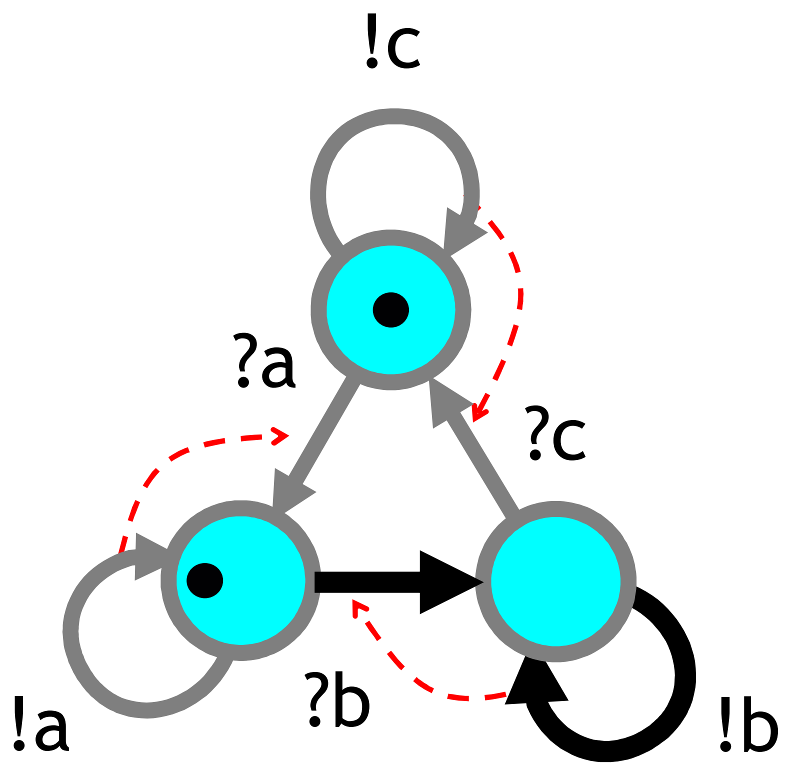
Example 3. Does it halt?

4 Automata



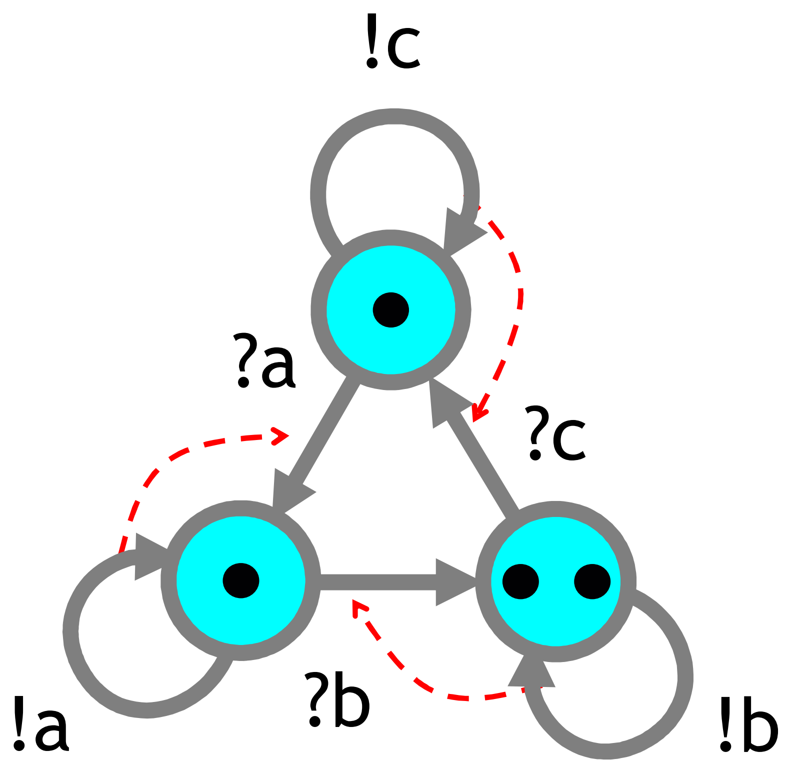
Example 3. Does it halt?

4 Automata



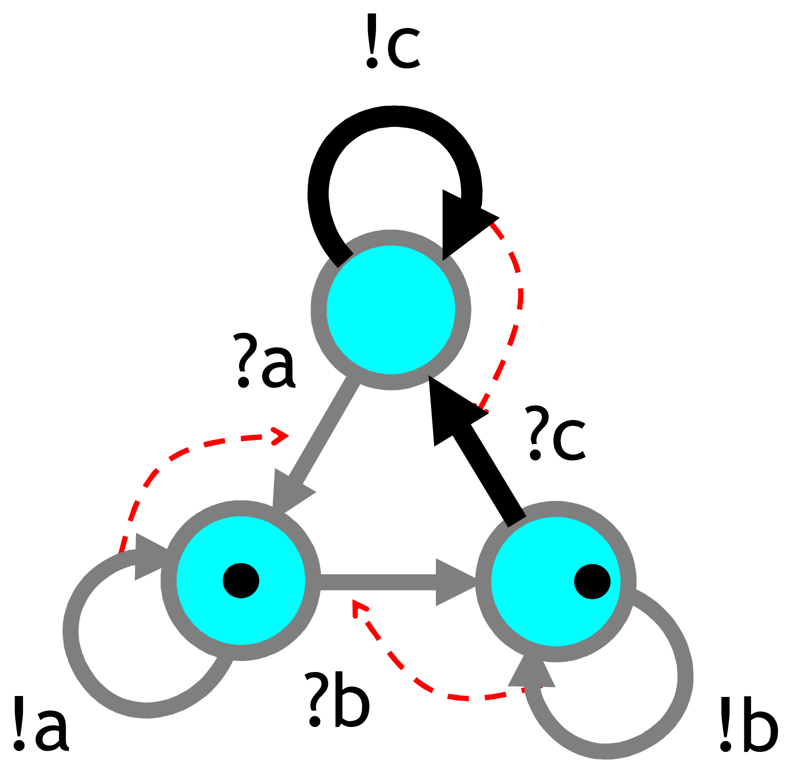
Example 3. Does it halt?

4 Automata



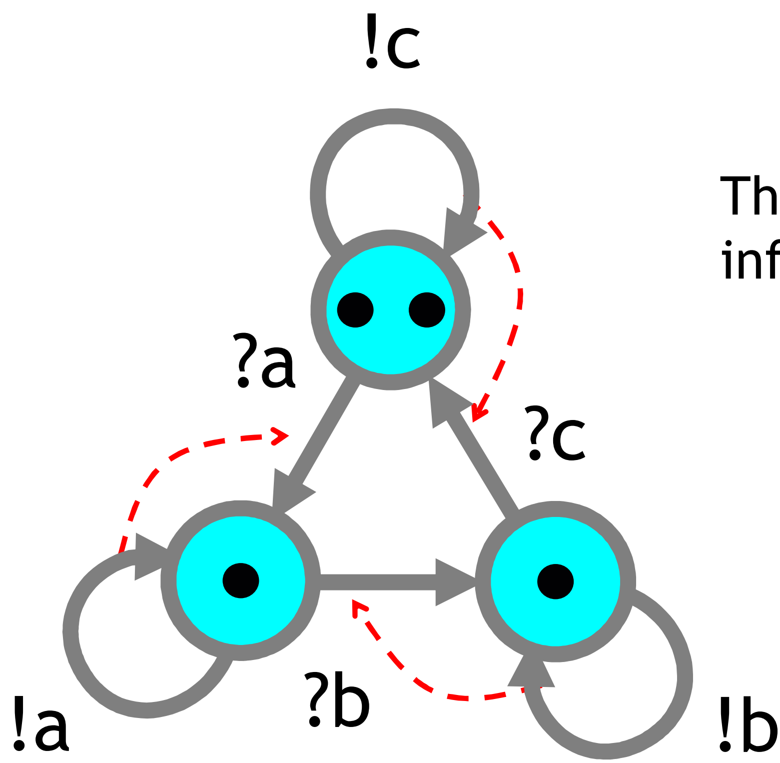
Example 3. Does it halt?

4 Automata



Example 3. Does it halt?

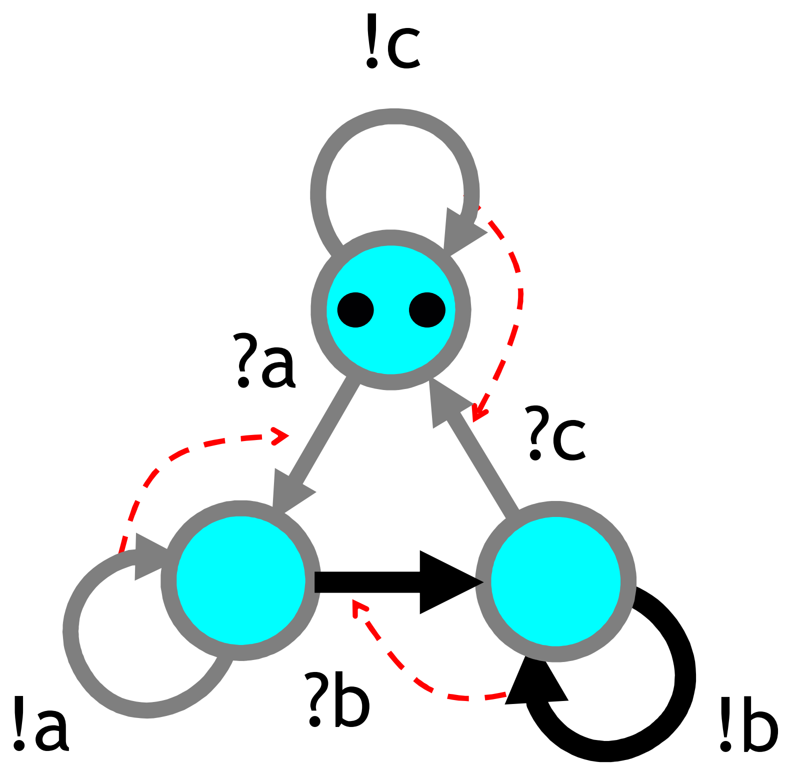
4 Automata



There is an infinite trace.

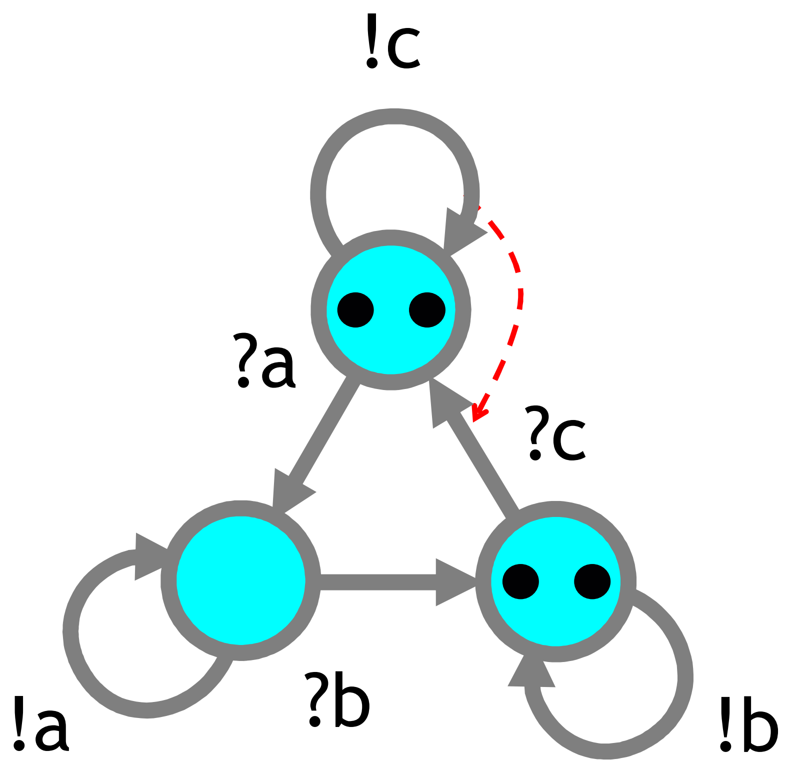
Example 3. Does it halt?

4 Automata



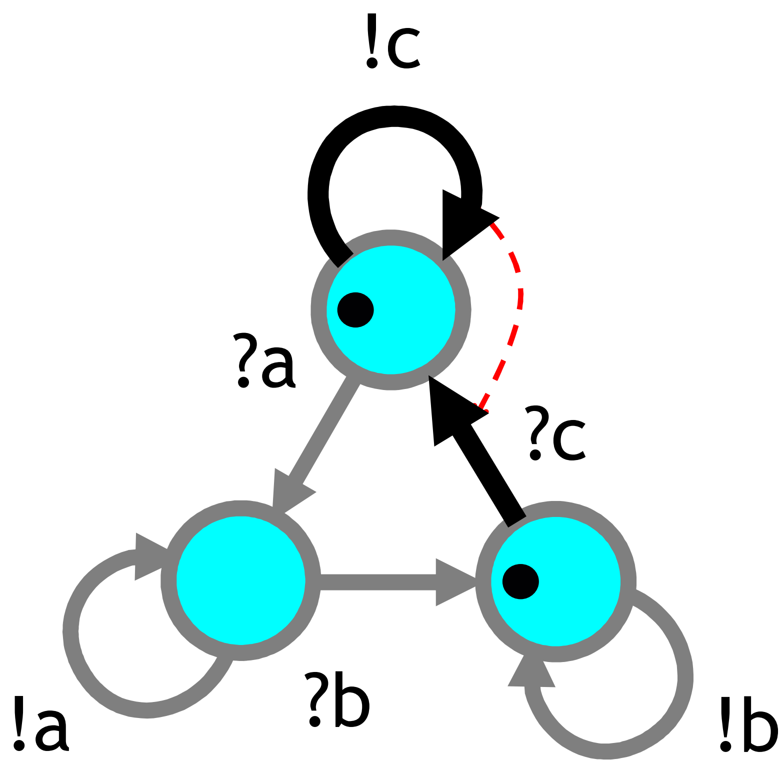
Example 3. Does it halt?

4 Automata



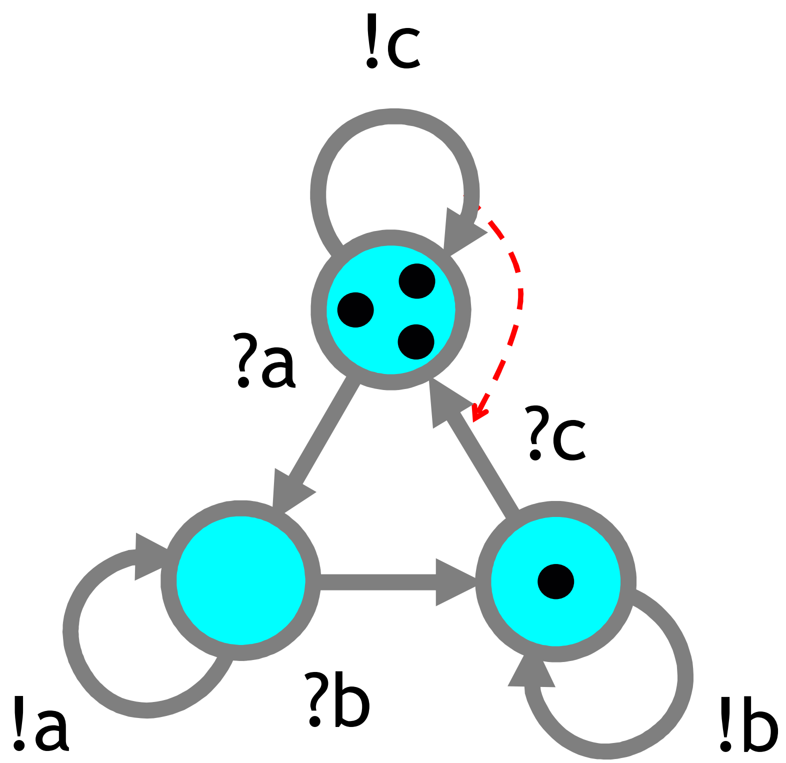
Example 3. Does it halt?

4 Automata



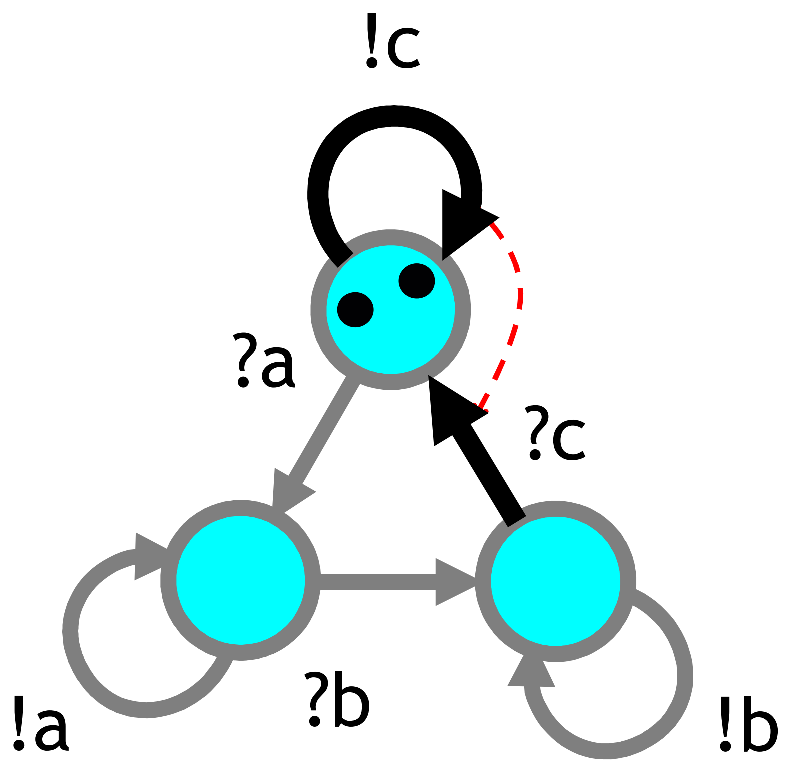
Example 3. Does it halt?

4 Automata



Example 3. Does it halt?

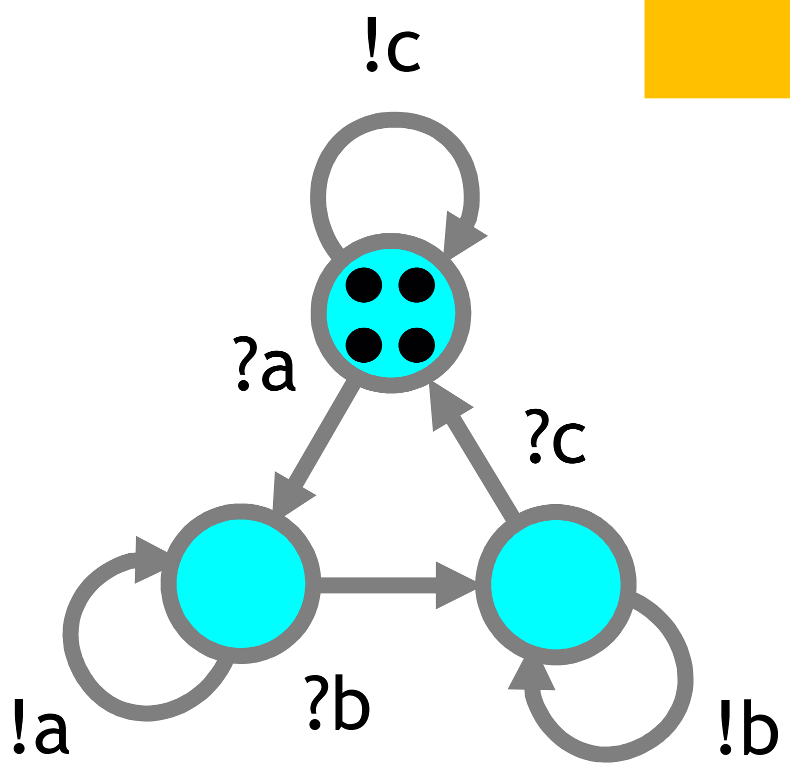
4 Automata



Example 3. Does it halt?

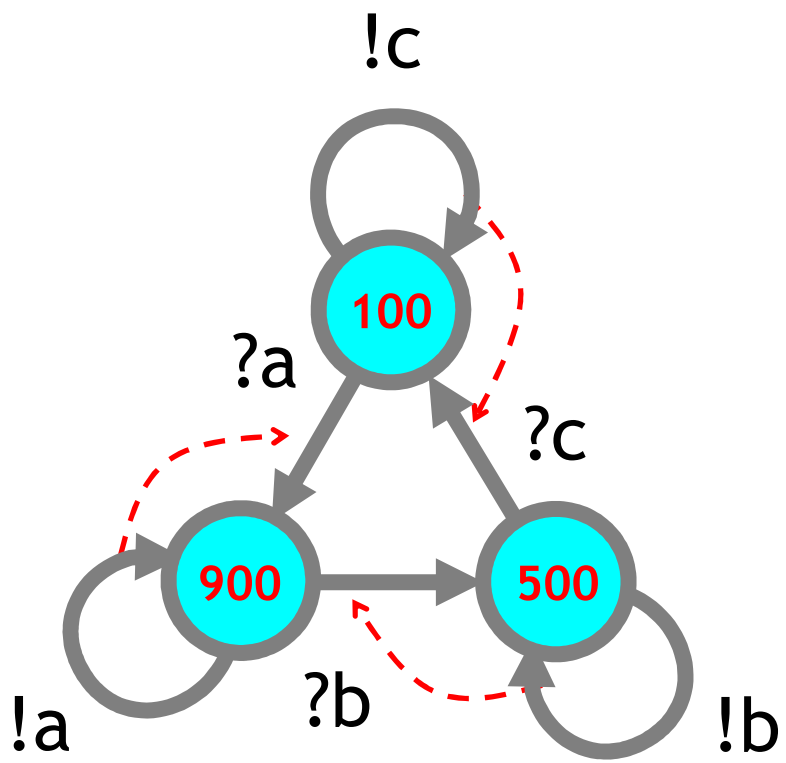
4 Automata

YES



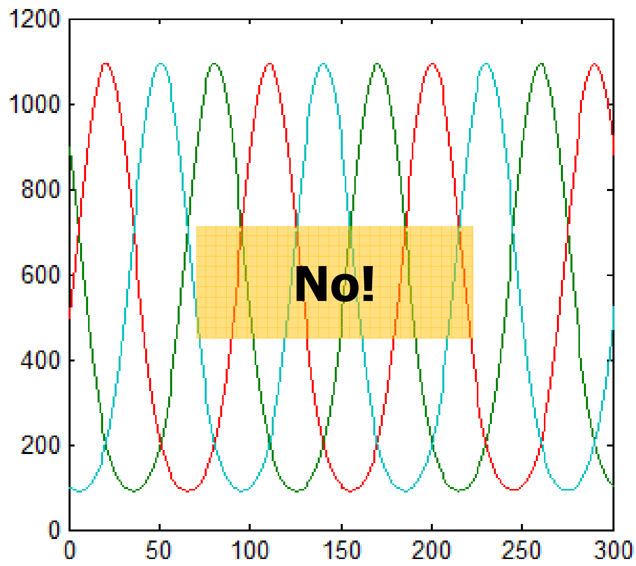
Example 3. Does it halt?

1500 Automata



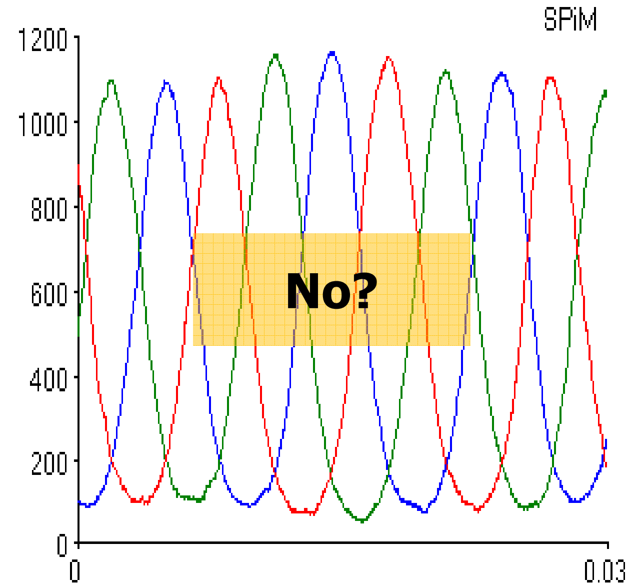
Example 3. Does it halt?

"Experimental Evidence"



Continuous-State Simulation

```
interval/step [0:0.0001:0.03]
(A)   dx1/dt = - x1*x2 + x3*x1      900.0
(B)   dx2/dt = - x2*x3 + x1*x2      500.0
(C)   dx3/dt = - x3*x1 + x2*x3      100.0
```



Discrete-State Simulation

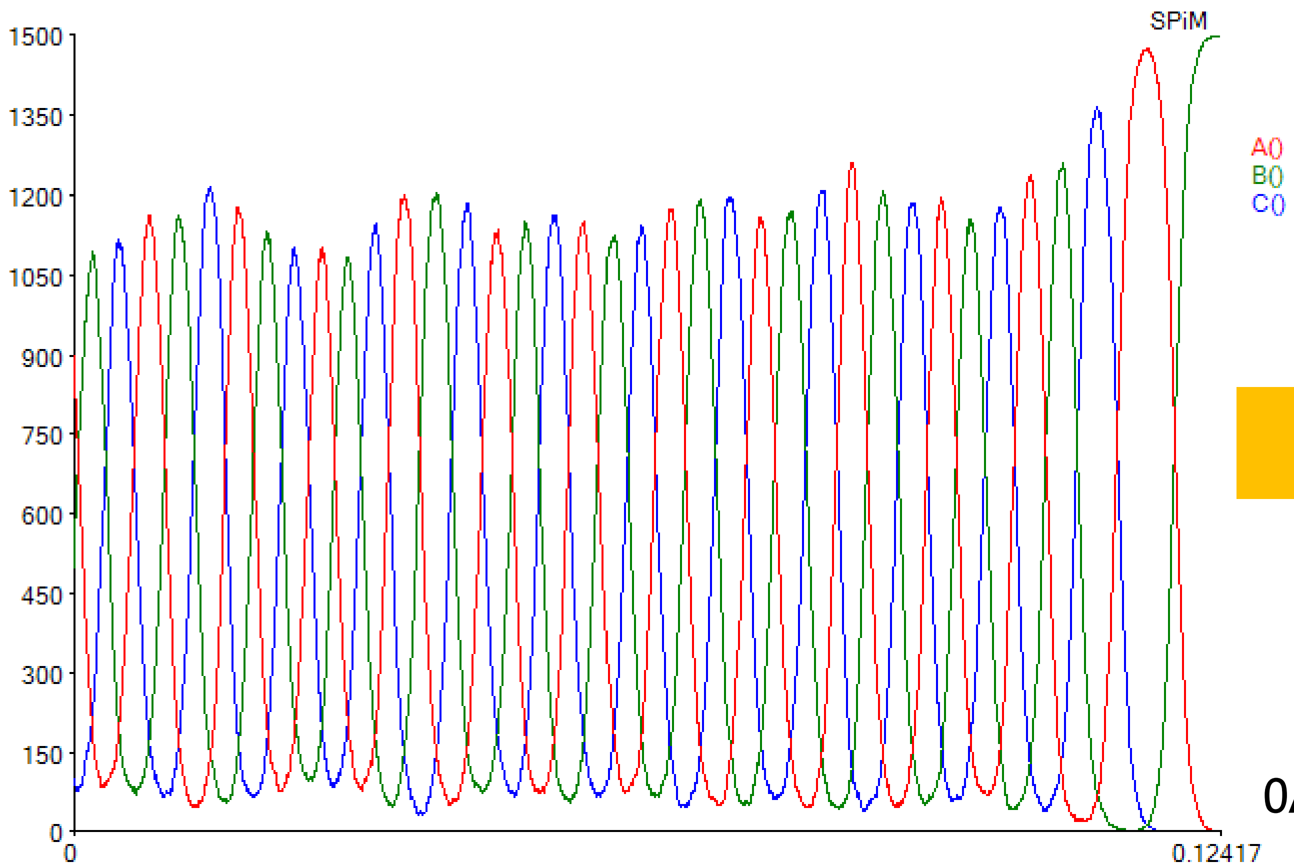
```
directive sample 0.03 1000
directive plot A(); B(); C()

new a@1.0:chan new b@1.0:chan new c@1.0:chan
let A() = do !a;A() or ?b; B()
and B() = do !b;B() or ?c; C()
and C() = do !c;C() or ?a; A()

run (900 of A() | 500 of B() | 100 of C())
```

Example 3. Does it halt?

But in a longer experiment...

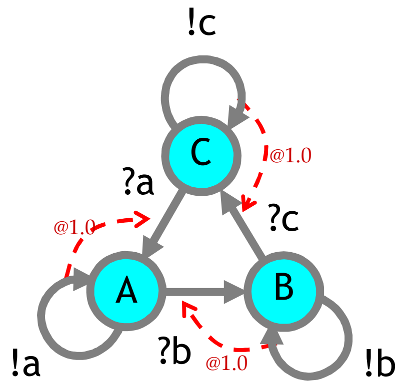


Discrete-State Simulation

YES

0A + 1500B + 0C

Example 3. Does it halt?



Termination strategy

It *can* terminate. (Apply reaction b until no more A's, then apply reaction c until no more B's. Then all are C.)

Nondeterministic termination

It *may* diverge (with 4+ molecules).

Stochastic termination

The probability measure of the terminated states of the oscillator's CMTC is 1.

=> Stochastic fairness

It *cannot* diverge!

Chemical Ground Form

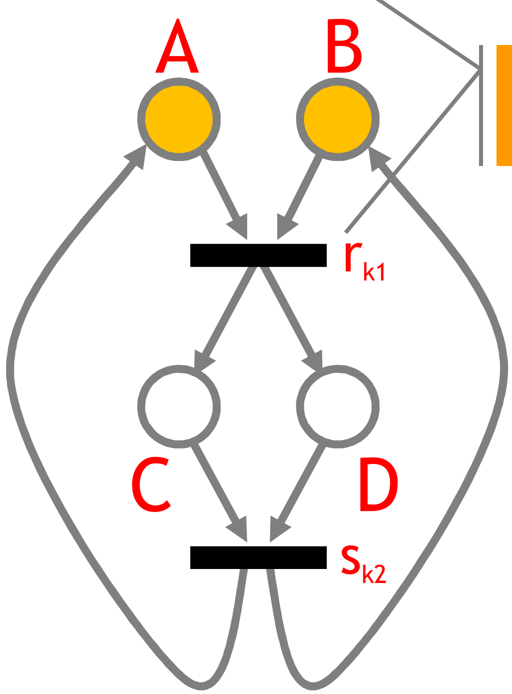
Chemistry vs. Automata

A process algebra (chemistry)

$r: A + B \rightarrow_{k1} C + D$

$s: C + D \rightarrow_{k2} A + B$

Does A become C or D?

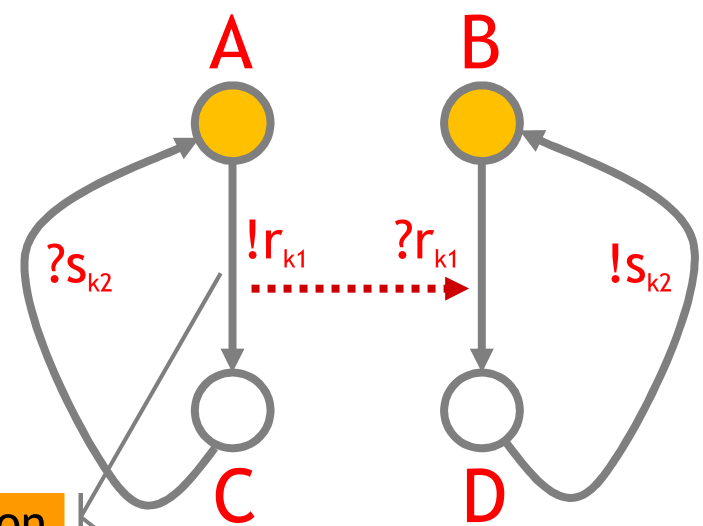


Reaction oriented

1 line per reaction

A Petri-Net-like representation. Precise and dynamic, but not modular, scalable, or maintainable.

A different process algebra (automata)



Interaction oriented

1 line per component

$A = !r_{k1}; C$

$C = ?s_{k2}; A$

$B = ?r_{k1}; D$

$D = !s_{k2}; B$

A becomes C not D!

The same "model"

Maps to a CTMC	Maps to a CTMC

A compositional graphical representation (precise, dynamic *and* modular) and the corresponding calculus.

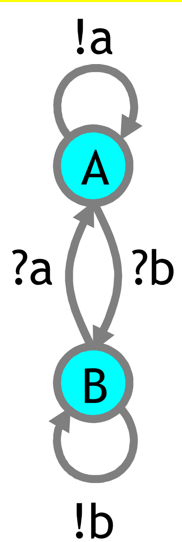
Chemical Ground Form (CGF)

$E ::= 0 \;\vdots\; X=M, E$	Reagents	
$M ::= 0 \;\vdots\; \pi;P \oplus M$	Molecules	
$P ::= 0 \;\vdots\; X \mid P$	Solutions	
$\pi ::= \tau_{(r)} \;\vdots\; ?a_{(r)} \;\vdots\; !a_{(r)}$	Actions (delay, input, output)	
$CGF ::= E,P$	Reagents plus Initial Conditions	

A stochastic subset of CCS
(no values, no restriction)

(To translate chemistry to processes we need a bit more than interacting automata: we may have "+" on the right of ®, that is we may need "|" after p.)

$\oplus$ is stochastic choice (vs. + for chemical reactions)
0 is the null solution ($P|0 = 0|P = P$)
  and null molecule ($M\oplus0 = 0\oplus M = M$)
Each X in E is a distinct *species*
Each name a is assigned a fixed rate r: $a_{(r)}$



Ex: Interacting Automata

(= finite-control CGFs: they use "|" only in initial conditions):

$A = !a;A \oplus ?b;B$ — Automaton in state A

$B = !b;B \oplus ?a;A$ — Automaton in state B

$A|A|B|B$ — Initial conditions: 2A and 2B

Finite Stochastic Reaction Networks

$A \to^r B_1 + \ldots + B_n$ (n≥0)	Unary Reaction	$d[A]/dt = -r[A]$	Exponential Decay
$A_1 + A_2 \to^r B_1 + \ldots + B_n$ (n≥0)	Hetero Reaction	$d[A_i]/dt = -r[A_1][A_2]$	Mass Action Law
$A + A \to^r B_1 + \ldots + B_n$ (n≥0)	Homeo Reaction	$d[A]/dt = -2r[A]^2$	Mass Action Law

(assuming $A \neq B_i \neq A_j$ for all i,j)

No other reactions!

JOURNAL OF CHEMICAL PHYSICS VOLUME 113, NUMBER 1

The chemical Langevin equation

Daniel T. Gillespie[a)]
Research Department, Code 4T4100D, Naval Air Warfare Center, China Lake, California 93555

Genuinely *trimolecular* reactions do not physically occur in dilute fluids with any appreciable frequency. *Apparently* trimolecular reactions in a fluid are usually the combined result of two bimolecular reactions and one monomolecular reaction, and involve an additional short-lived species.

Chapter IV: Chemical Kinetics
[David A. Reckhow , CEE 572 Course]
... reactions may be either elementary or non-elementary. Elementary reactions are those reactions that occur exactly as they are written, without any intermediate steps. These reactions almost always involve just one or two reactants. ... Non-elementary reactions involve a series of two or more elementary reactions. Many complex environmental reactions are non-elementary. In general, reactions with an overall reaction order greater than two, or reactions with some non-integer reaction order are non-elementary.

THE COLLISION THEORY OF REACTION RATES www.chemguide.co.uk
The chances of all this happening if your reaction needed a collision involving more than 2 particles are remote. All three (or more) particles would have to arrive at exactly the same point in space at the same time, with everything lined up exactly right, and having enough energy to react. That's not likely to happen very often!

Trimolecular reactions:

$A + B + C \to^r D$

the measured "r" is an (imperfect) aggregate of e.g.:

$A + B \leftrightarrow AB$

$AB + C \to D$

Enzymatic reactions:

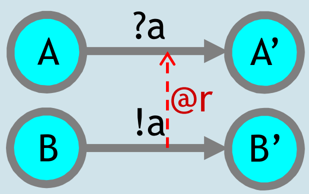
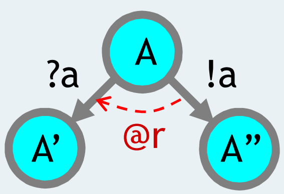
$S \xrightarrow{E}{}^r P$

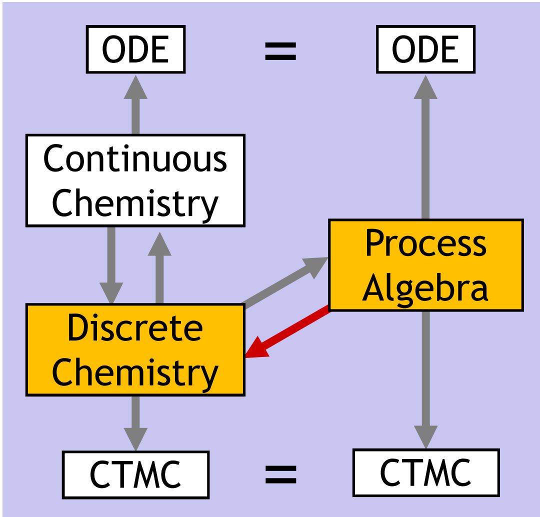
the "r" is given by Michaelis-Menten (approximated steady-state) laws:

$E + S \leftrightarrow ES$

$ES \to P + E$

From CGF to FSRN (by example)

Interacting Automata →	Discrete Chemistry
initial states A | A | … | A	initial quantities $\#A_0$
	$A \xrightarrow{r} A'$
	$A+B \xrightarrow{r} A'+B'$
	$A+A \xrightarrow{2r} A'+A''$



From FSRN to CGF (by example)

v_1: $A+B \rightarrow_{k_1} C+C$
v_2: $A+C \rightarrow_{k_2} D$
v_3: $C \rightarrow_{k_3} E+F$
v_4: $F+F \rightarrow_{k_4} B$

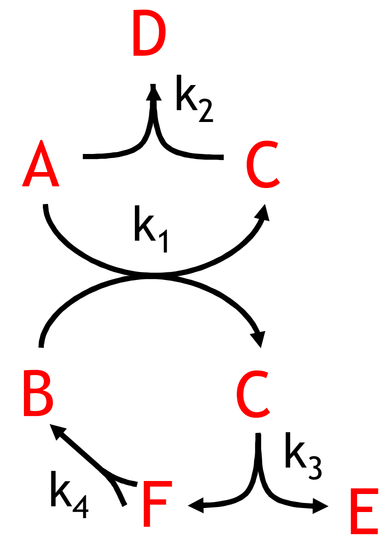
Interaction Matrix

channels and rates (1 per reaction)

Half-rate for homeo reactions

definitions (1 per species)

	$v_{1(k1)}$	$v_{2(k2)}$	$v_{3(k3)}$	$v_{4(k4/2)}$
A	?;(C|C)	?;D		
B	!;0			
C		!;0	t;(E|F)	
D				
E				
F				?;B !;0

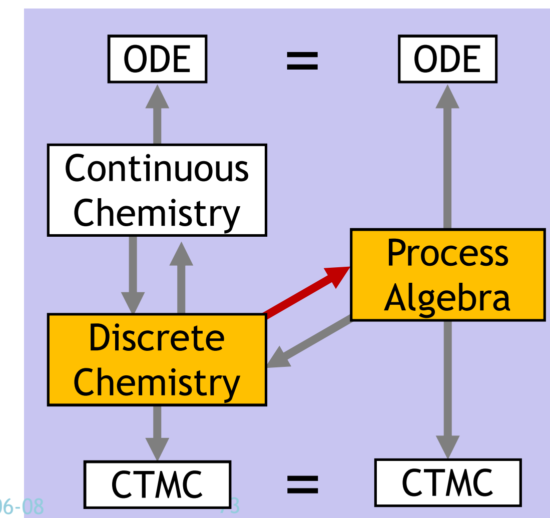


1: Fill the matrix by columns:

Degradation reaction v_i: $X \rightarrow_{k_i} P_i$
  add $t;P_i$ to $\langle X, v_{ij} \rangle$.
Hetero reaction v_i: $X+Y \rightarrow_{k_i} P_i$
  add $?;P_i$ to $\langle X, v_i \rangle$ and $!;0$ to $\langle Y, v_i \rangle$
Homeo reaction v_i: $X+X \rightarrow_{k_i} P_i$
  add $?;P_i$ and $!;0$ to $\langle X, v_i \rangle$

2: Read the result by rows:

$A = \ ?v_{1(k1)};(C|C) \oplus ?v_{2(k2)};D$
$B = \ !v_{1(k1)};0$
$C = \ !v_{2(k2)};0 \oplus t_{k3};(E|F)$
$D = 0$
$E = 0$
$F = \ ?v_{4(k4/2)};B \oplus !v_{4(k4/2)};0$



Discrete Semantics of FSRN

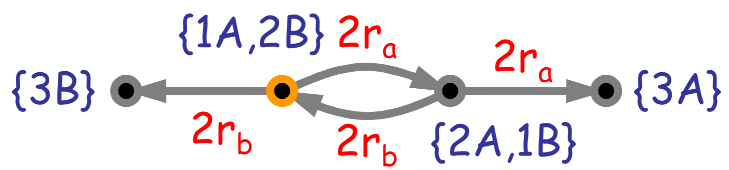
Syntax:

A+B $\to^r$ A+A

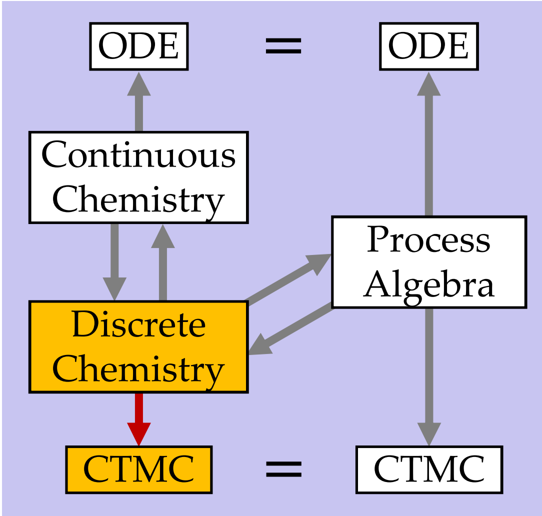
A+B $\to^r$ B+B

A+B+B

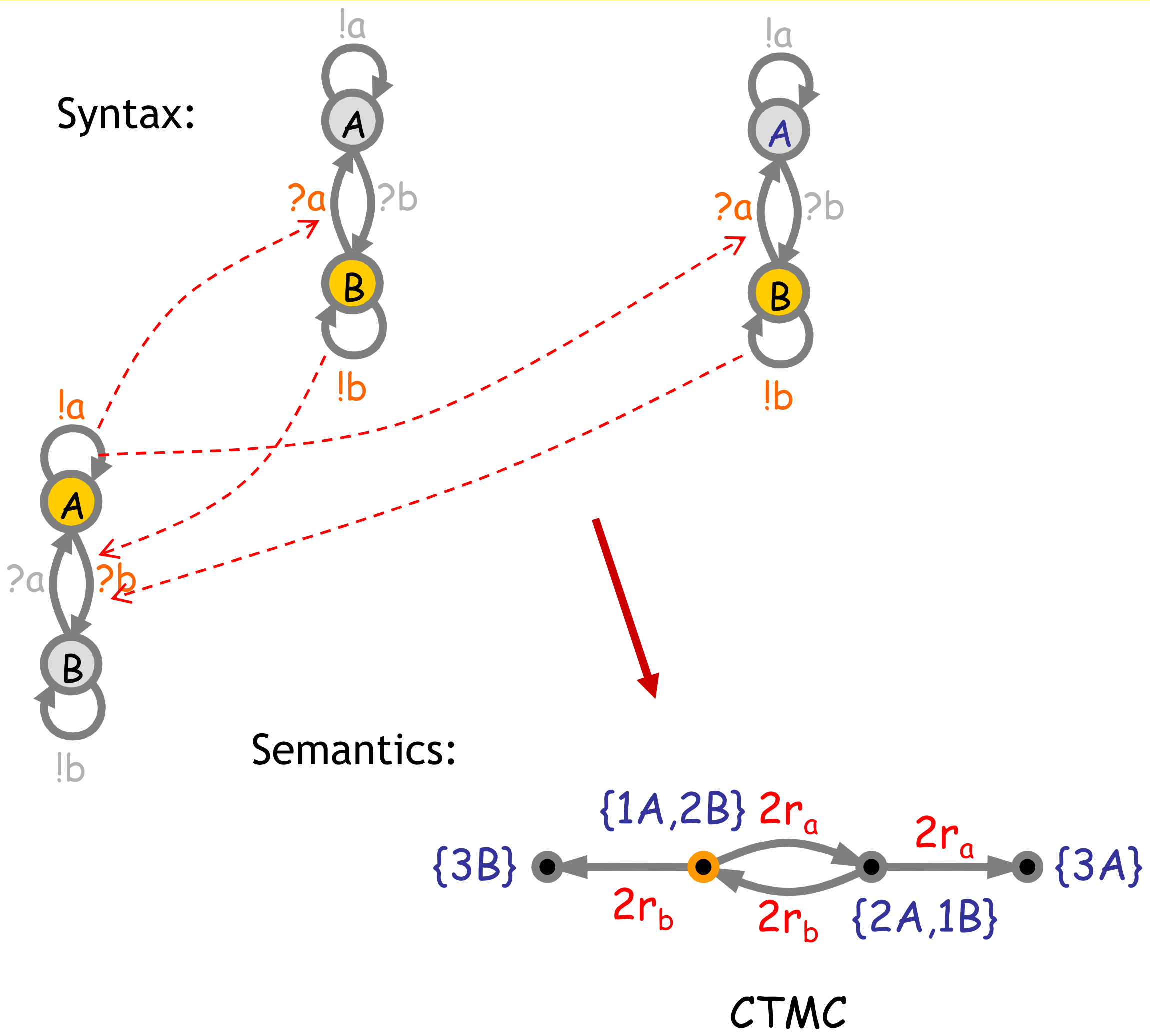
Semantics:

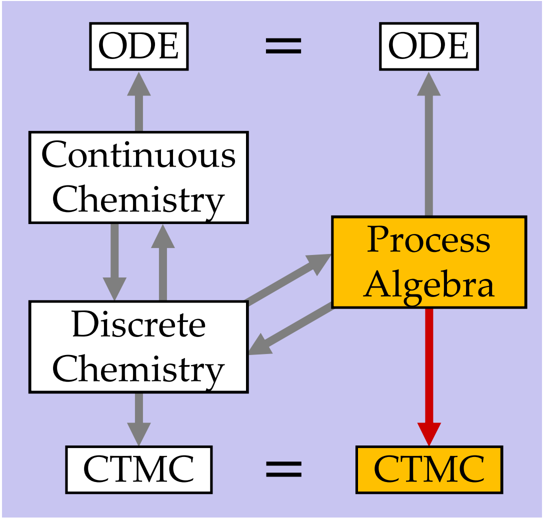


CTMC



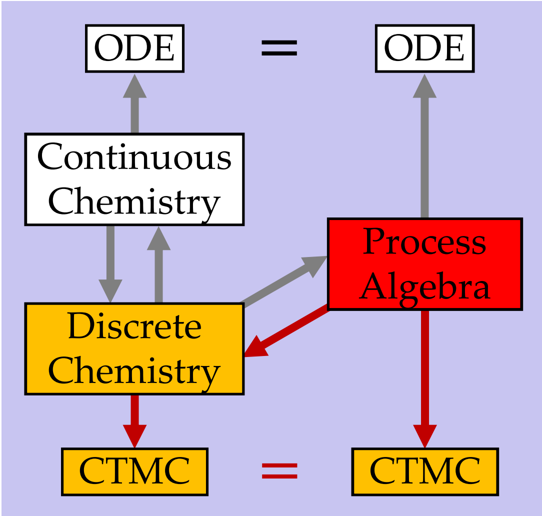
Discrete Semantics of CGF

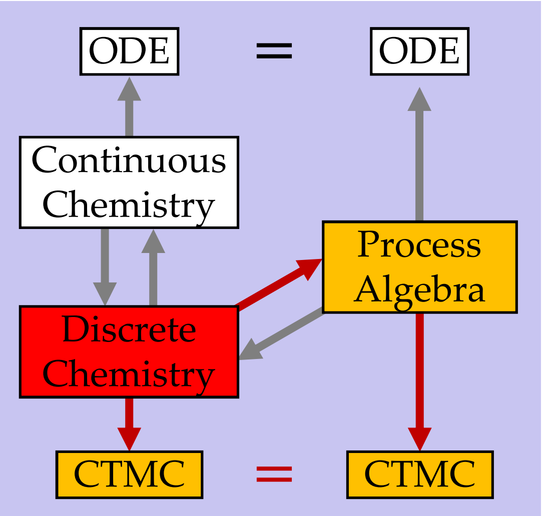




Discrete State Equivalence

- Def: ≈ is equivalent CTMC's (isomorphic graphs with same rates).
- Thm: E ≈ Ch(E)
- Thm: C ≈ Pi(C)





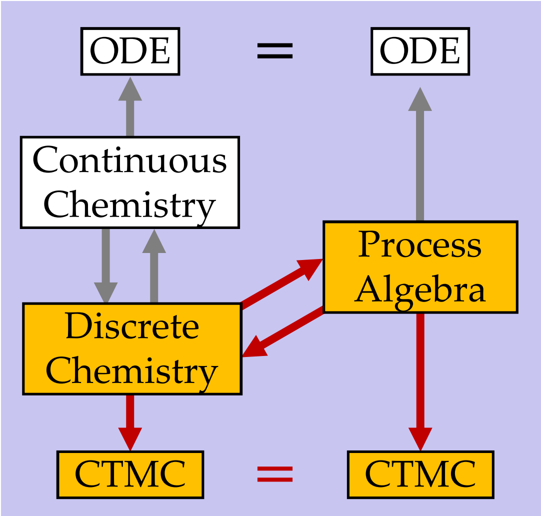
- For each E there is an E' ≈ E that is detangled (E' = Pi(Ch(E)))
- For each E in automata form there is an an E' ≈ E that is detangled and in automata form (E' = Detangle(E)).

CGF = FSRN

This is enough to establish that the process algebra is really faithful to the chemistry.

But CTMC are not the "ultimate semantics" because there are still questions of when two different CTMCs are actually equivalent (e.g. "lumping").

The "ultimate semantics" of chemistry is the *Chemical Master Equation* (derivable from the Chapman-Kolmogorov equation of the CTMC).

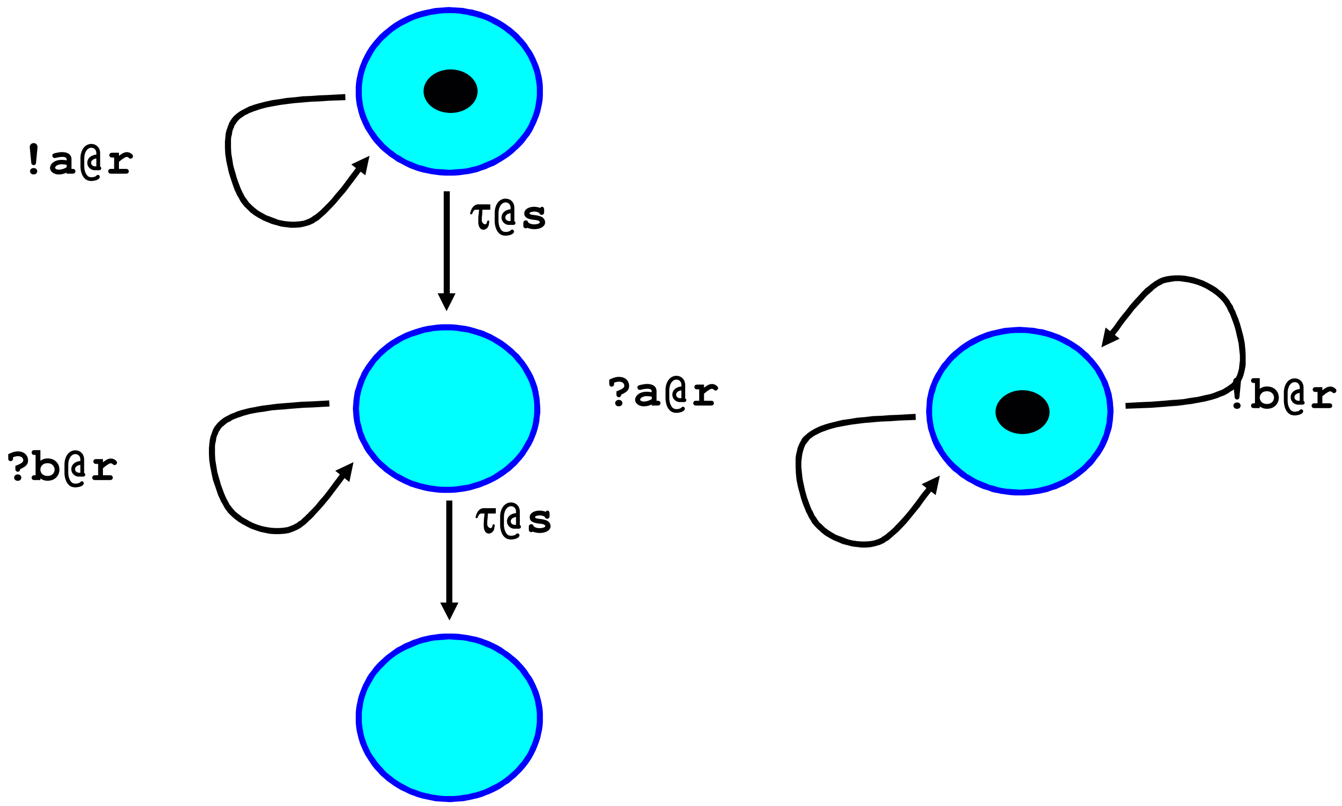


But it's all just Petri Nets!

- It is possible to translate an arbitrary CGF or FSRN into a Place/Transition Petri Net.
  - Ignoring rates, and of course loosing compositionality.
- Pretty much everything is decidable in P/T Nets.
  - In particular, reachability of a dead state.
- Hence both CGF and FSRN are not Turing-complete!
  - Basic chemistry can't compute!
    (Soloveichik et. al., Natural Computing 2008)
  - Even though stochastic chemistry is extremely rich, e.g. including chaotic systems.

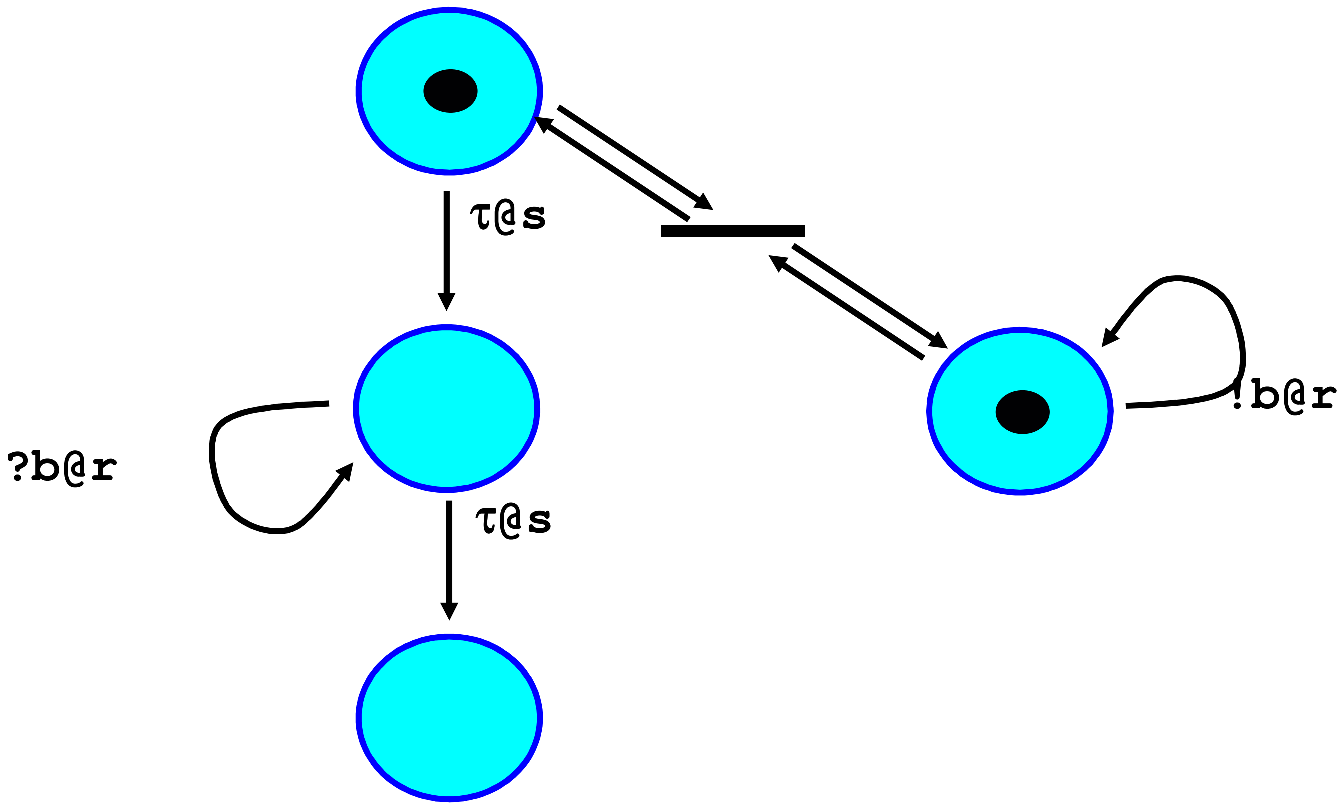
A Petri net semantics for CGF

- One place for each Species
- One transition for each reaction



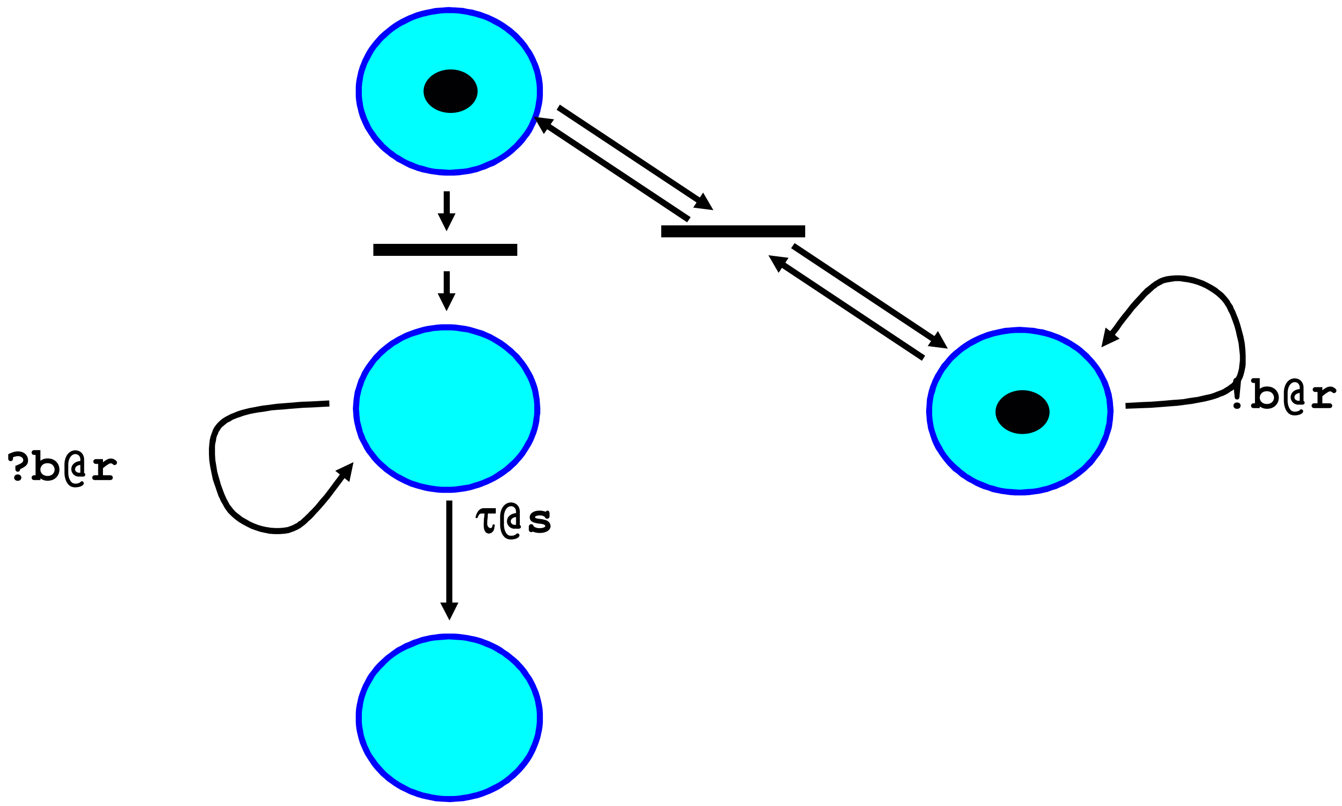
A Petri net semantics for CGF

- One place for each Species
- One transition for each reaction



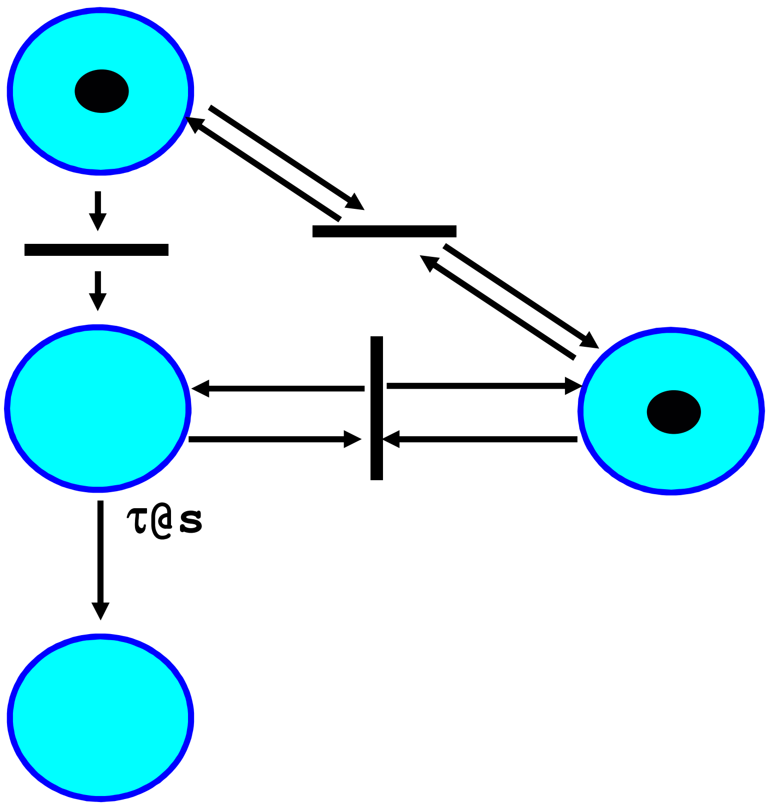
A Petri net semantics for CGF

- One place for each Species
- One transition for each reaction



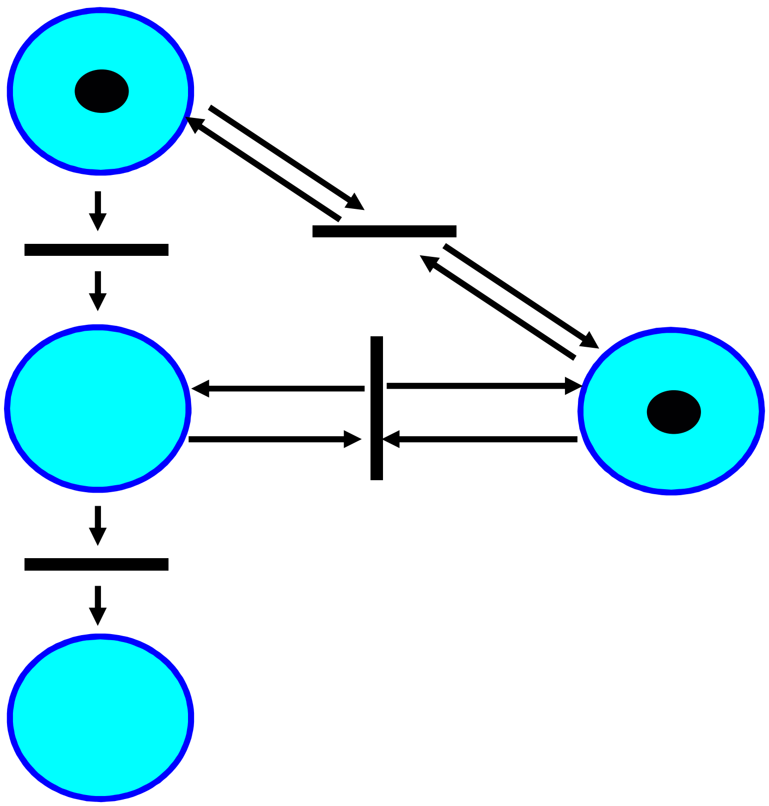
A Petri net semantics for CGF

- One place for each Species
- One transition for each reaction



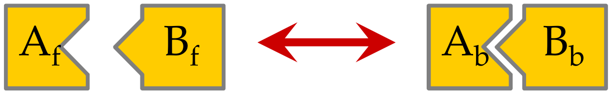
A Petri net semantics for CGF

- One place for each Species
- One transition for each reaction

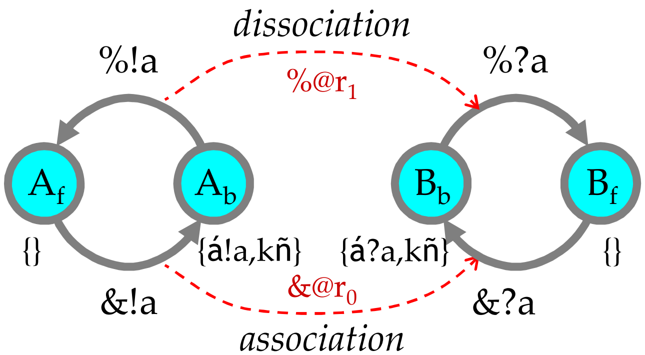


Biochemical Ground Form

Biochemistry = Interaction + Complexation



- Complexation is what proteins "do", in contrast to simpler chemicals.



- Leading to a process algebra that we call the Biochemical Ground Form (BGF).

Biochemical Ground Form (BGF)

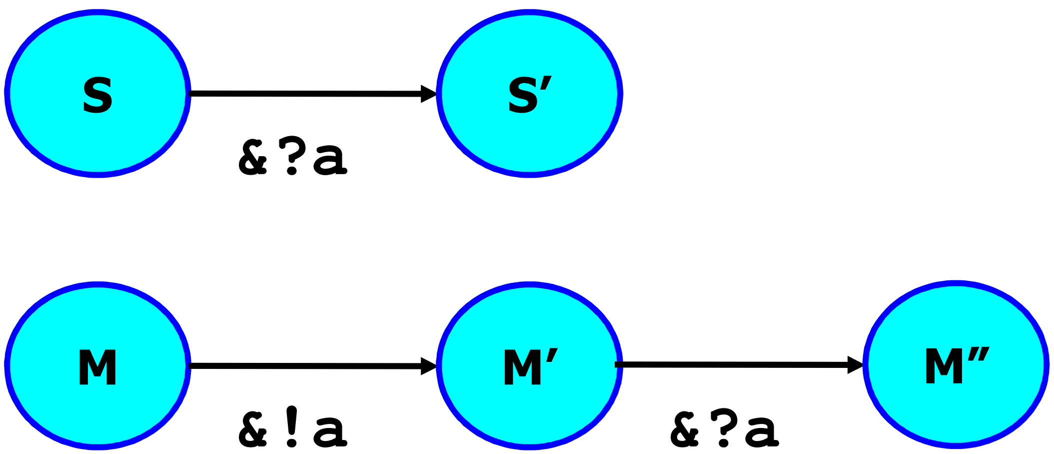
A stochastic subset of π
(no values, implicit restriction)

E ::= 0 ⋮ X=M, E	Reagents
M ::= 0 ⋮ π;P ⊕ M	Molecules
P ::= 0 ⋮ X | P	Products
π ::= $\tau_{(r)}$ ⋮ $?a_{(r)}$ ⋮ $!a_{(r)}$	Actions (delay, input, output,
⋮ $\&?a_{(r)}$ ⋮ $\&!a_{(r)}$	association,
⋮ $\%?a_{(r)}$ ⋮ $\%!a_{(r)}$	dissociation)
S ::= 0 ⋮ X_H | S	Solutions
H ::= 0 ⋮ <?a,k>::H	Association Histories
⋮ <!a,k>::H	
BGF ::= E,S	Reagents plus Initial Solution

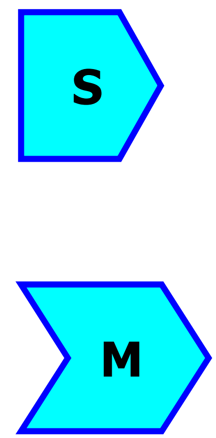
⊕ is stochastic choice (vs. + for chemical reactions)
0 is the null solution (P|0 = 0|P = P)
and null molecule (M⊕0 = 0⊕M = M)
Each X in E is a distinct *species*
Each name a is assigned a fixed rate r: $a_{(r)}$

Ex 1: Linear Polymerization

S = &?a; S’
M = &!a; M’
M’ = &?a; M”



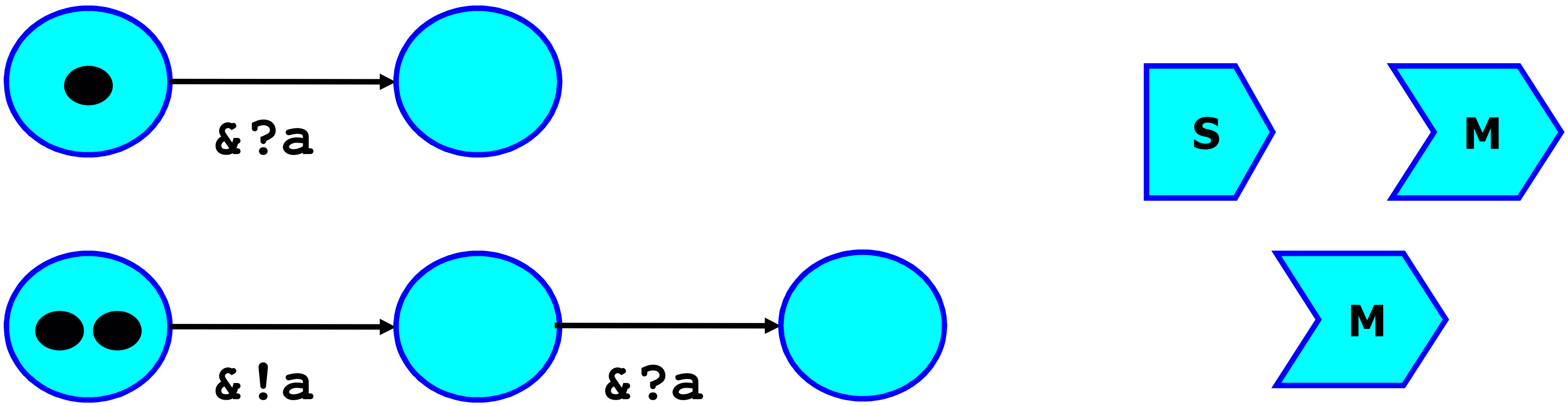
Seed



Monomer

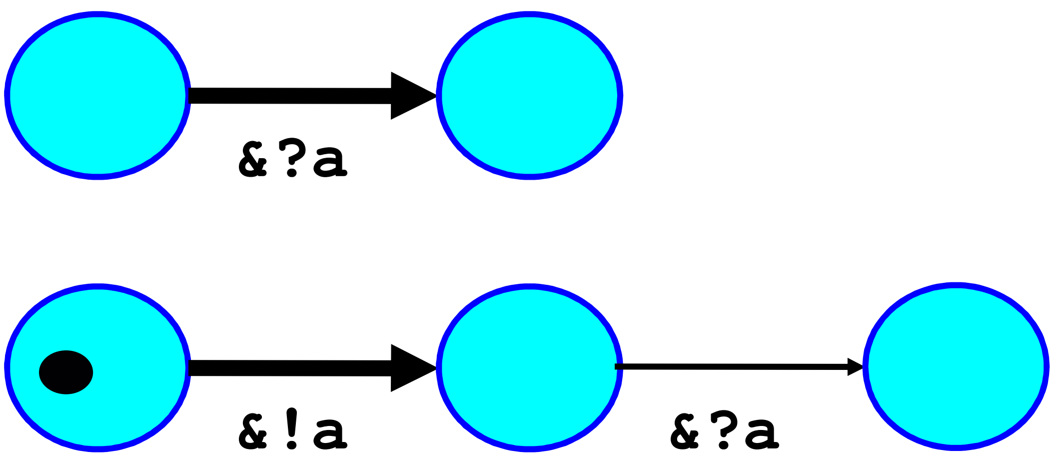
Ex 1: Linear Polymerization

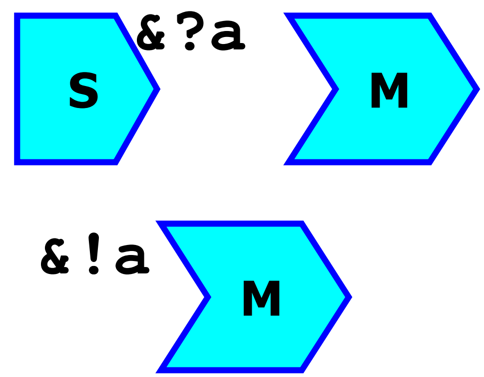
S = &?a; S'
M = &!a; M'
M' = &?a; M"



Ex 1: Linear Polymerization

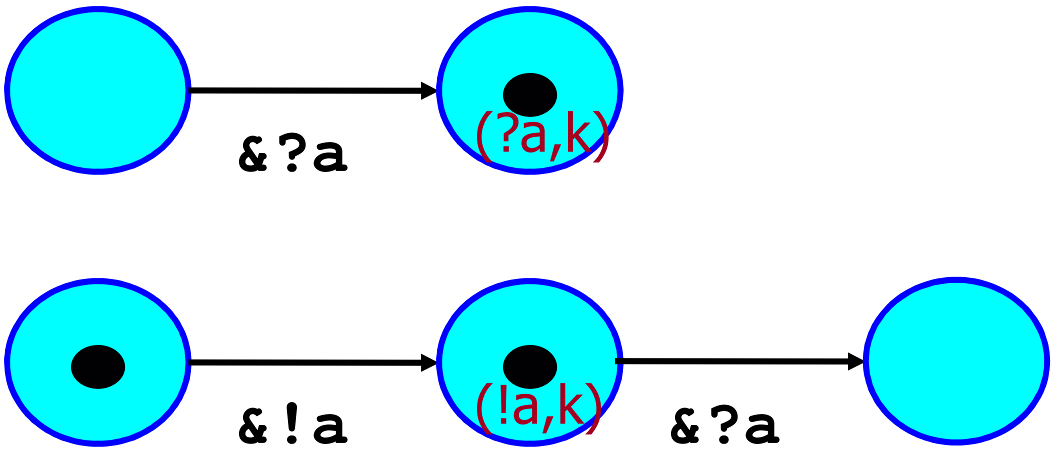
S = &?a; S’
M = &!a; M’
M’ = &?a; M”

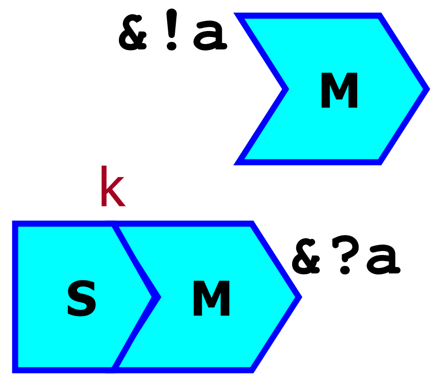




Ex 1: Linear Polymerization

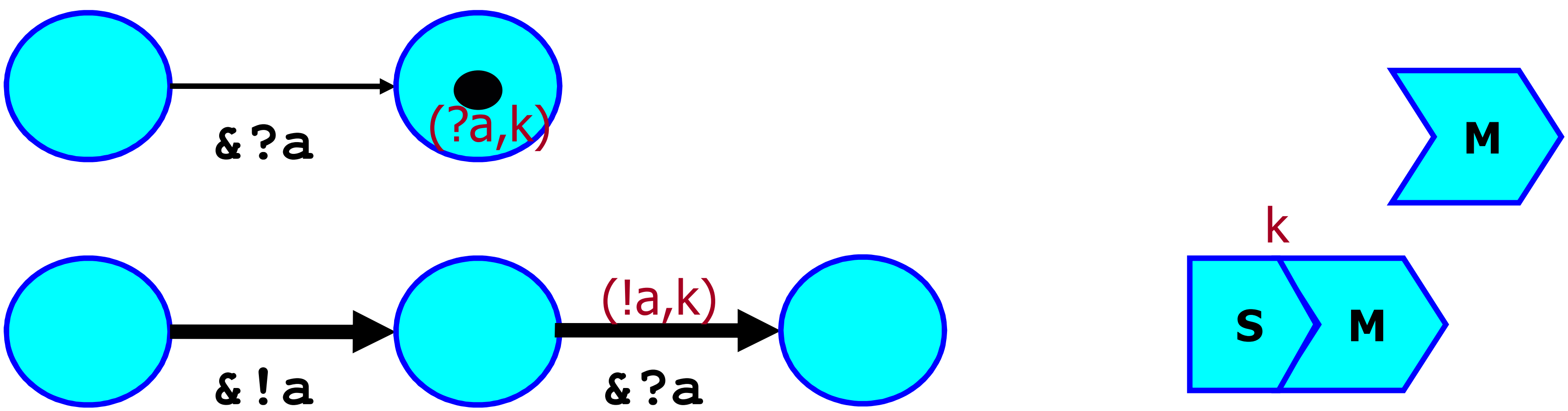
S = &?a; S'
M = &!a; M'
M' = &?a; M"





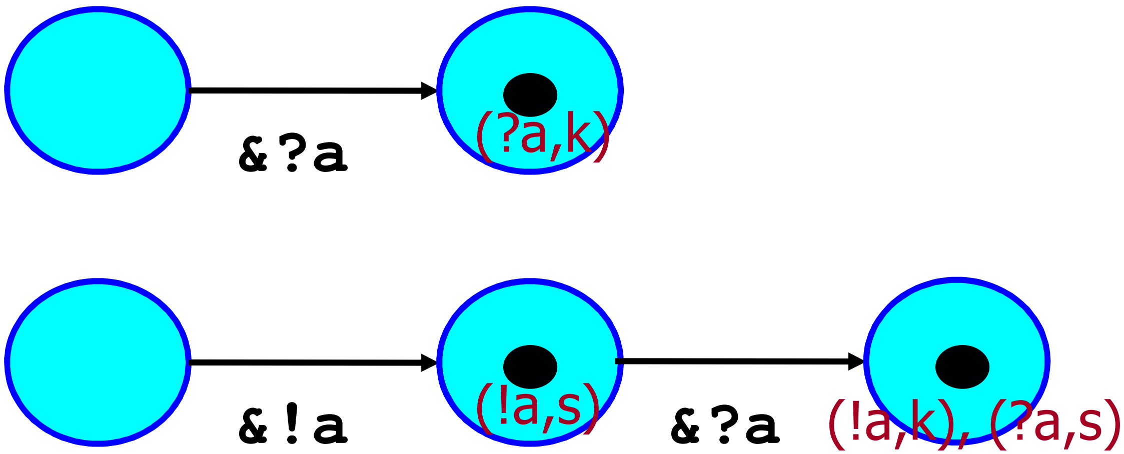
Ex 1: Linear Polymerization

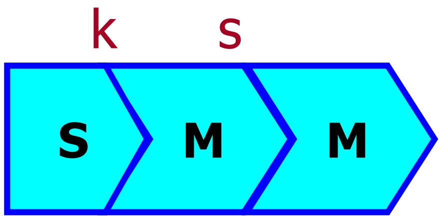
S = &?a; S'
M = &!a; M'
M' = &?a; M"



Ex 1: Linear Polymerization

S = &?a; S’
M = &!a; M’
M’ = &?a; M”

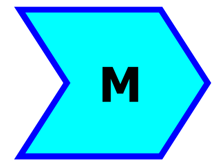


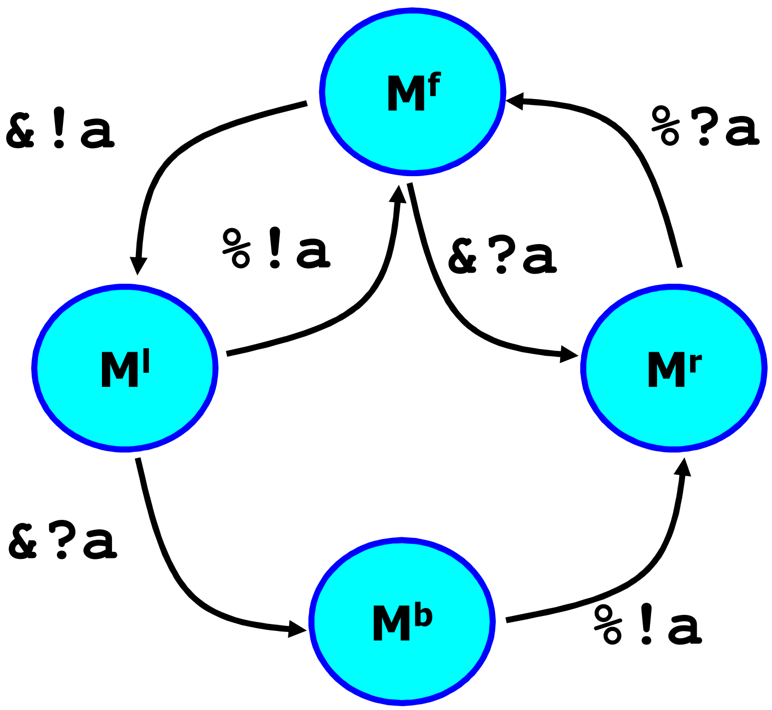


Ex 2: Actin Polymerization

$M^f = \&!a; M^l \oplus \&?a; M^r$
$M^l = \%!a; M^f \oplus \&?a; M^b$
$M^r = \%?a; M^f$
$M^b = \%!a; M^r$

Grows only to the right, shrinks only from the left





M^f = free on both sides
M^l = bound on the left
M^r = bound on the right
M^b = bound on both sides

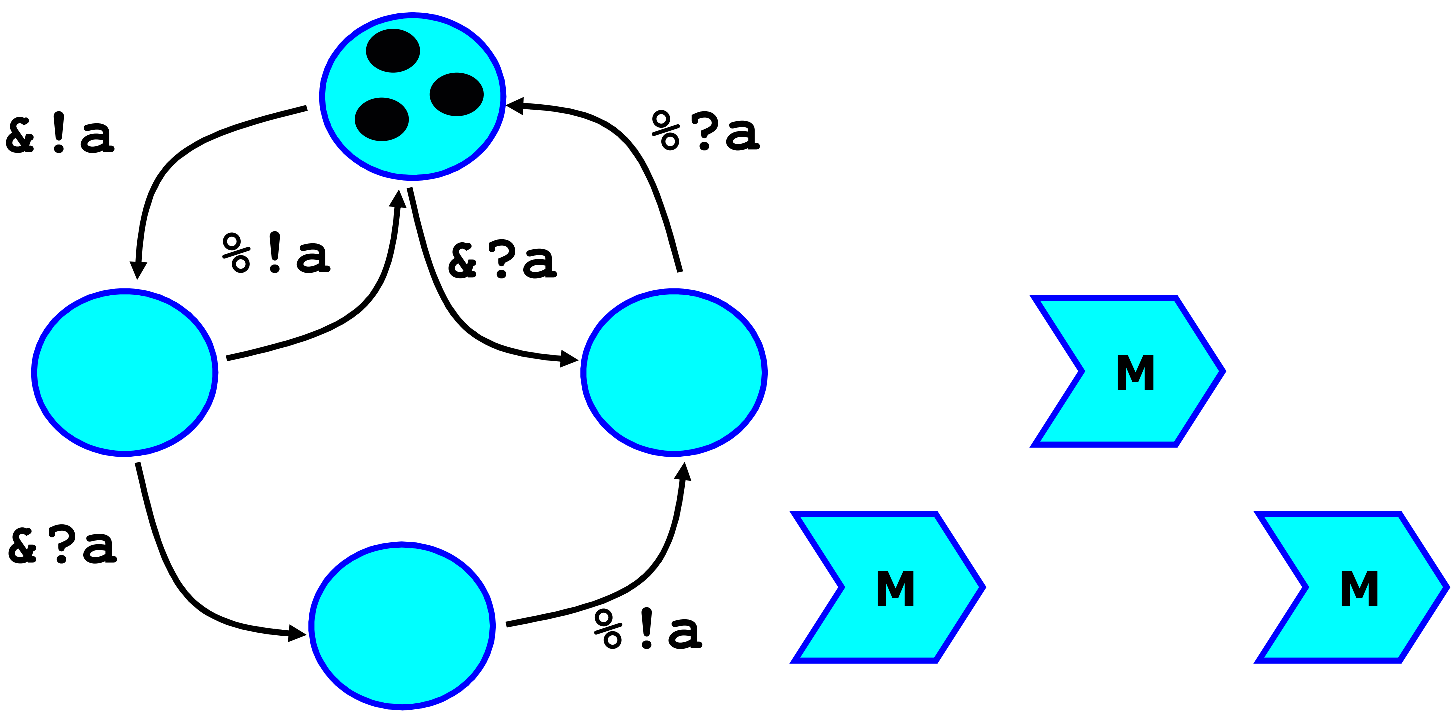
Ex 2: Actin Polymerization

M^f = &!a; M^l ⊕ &?a; M^r

M^l = %!a; M^f ⊕ &?a; M^b

M^r = %?a; M^f

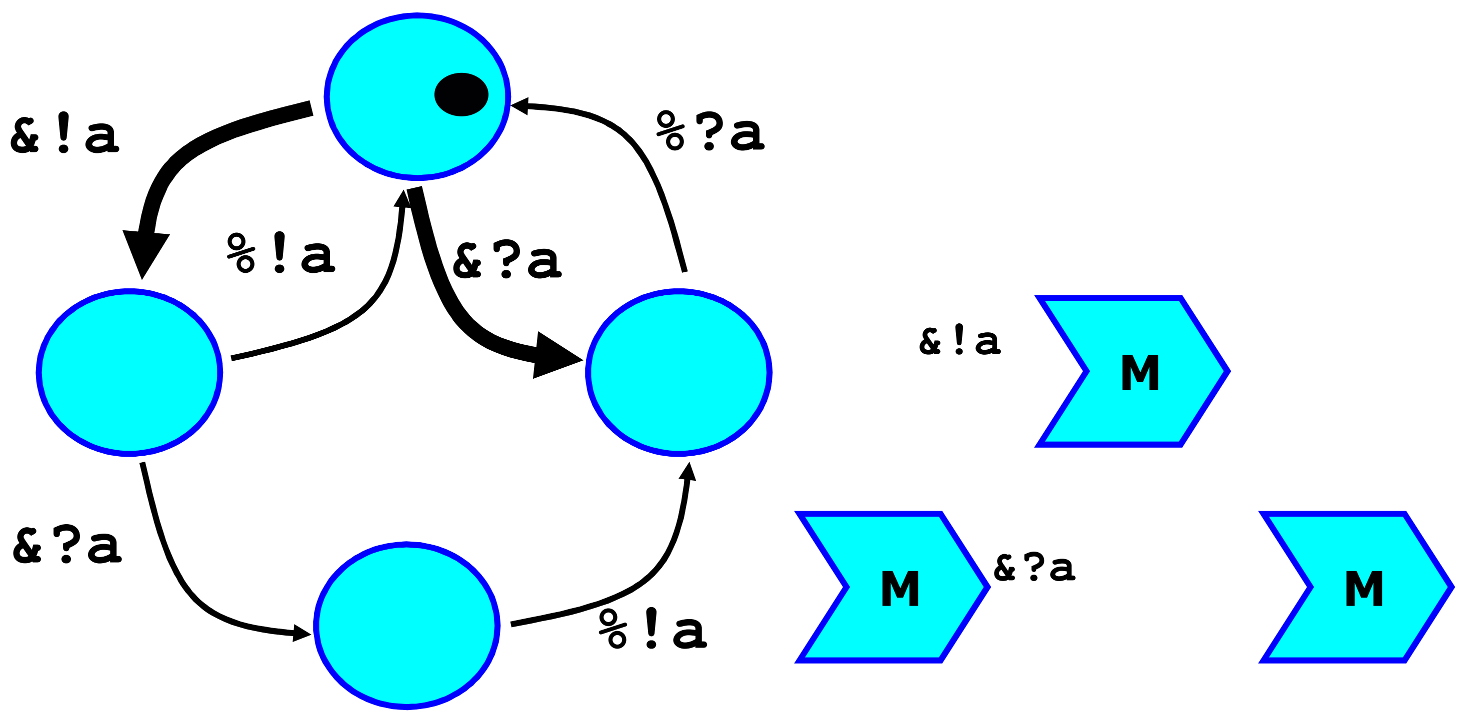
M^b = %!a; M^r



Ex 2: Actin Polymerization

- Each association has a unique key
  - Keys are stored in the molecule's history

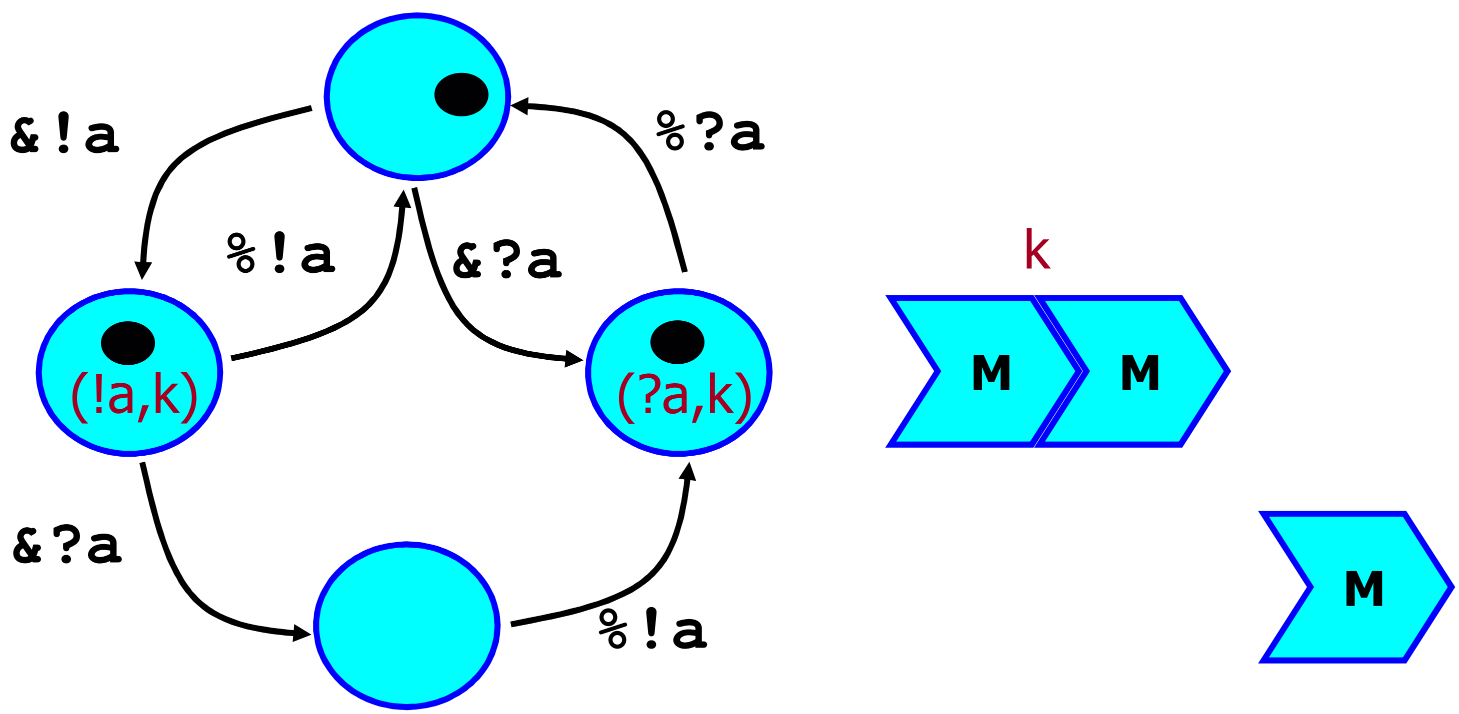
$$M^f = \&!a;\ M^l \oplus \&?a;\ M^r$$
$$M^l = \%!a;\ M^f \oplus \&?a;\ M^b$$
$$M^r = \%?a;\ M^f$$
$$M^b = \%!a;\ M^r$$



Ex 2: Actin Polymerization

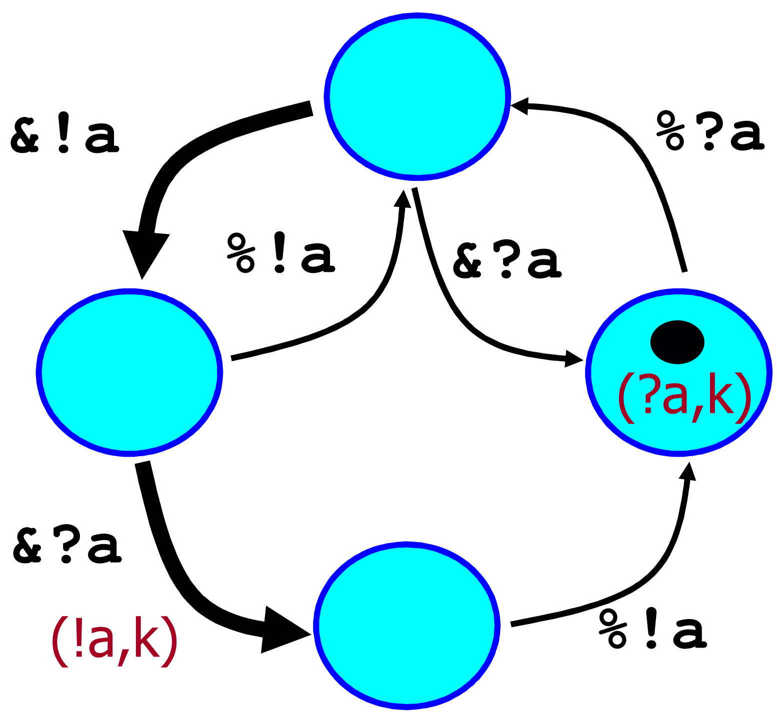
- Each association has a unique key
  - Keys are stored in the molecule's history

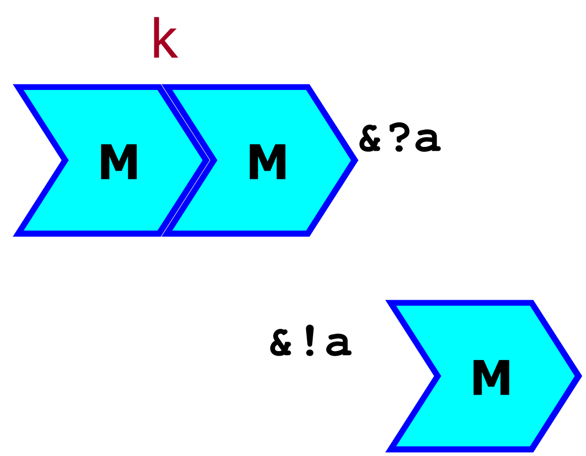
M^f = &!a; $M^l \oplus$ &?a; M^r
M^l = %!a; $M^f \oplus$ &?a; M^b
M^r = %?a; M^f
M^b = %!a; M^r



Ex 2: Actin Polymerization

$M^f = \&!a;\ M^l \oplus \&?a;\ M^r$
$M^l = \%!a;\ M^f \oplus \&?a;\ M^b$
$M^r = \%?a;\ M^f$
$M^b = \%!a;\ M^r$





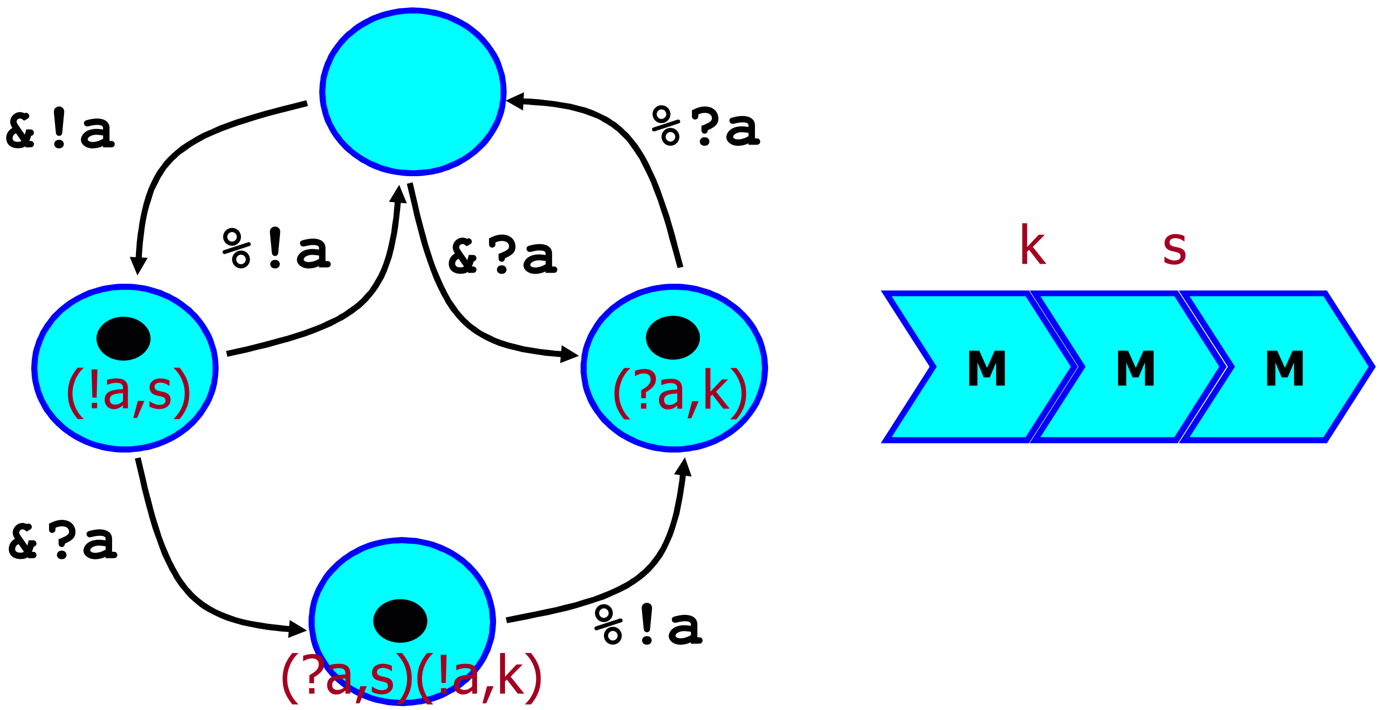
Ex 2: Actin Polymerization

$M^f = \&!a;\ M^l \oplus \&?a;\ M^r$

$M^l = \%!a;\ M^f \oplus \&?a;\ M^b$

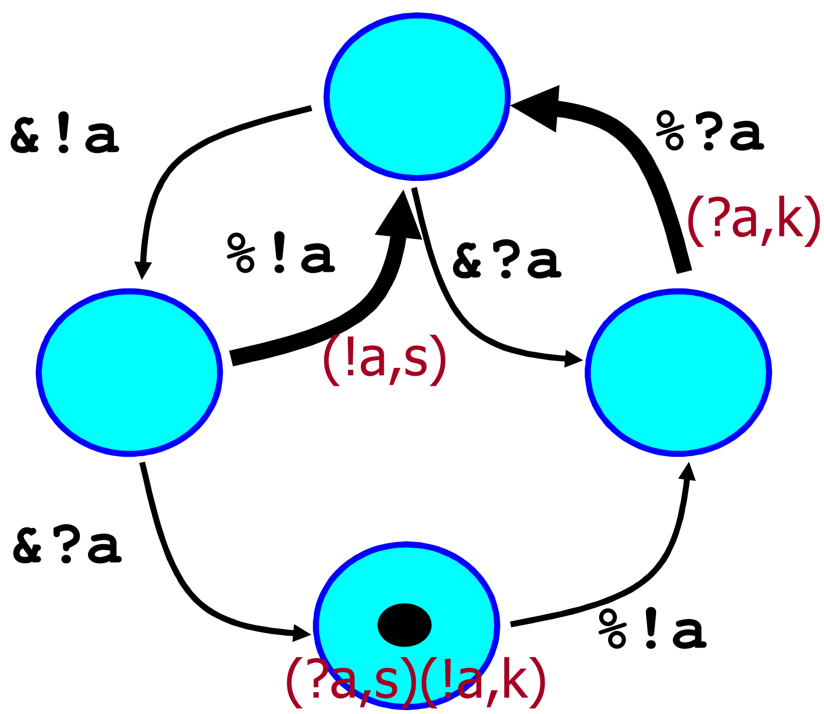
$M^r = \%?a;\ M^f$

$M^b = \%!a;\ M^r$

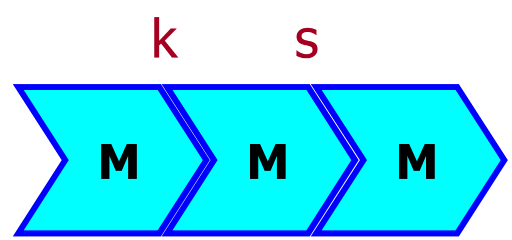


Ex 2: Actin Polymerization

$M^f = \&!a;\ M^l \oplus \&?a;\ M^r$
$M^l = \%!a;\ M^f \oplus \&?a;\ M^b$
$M^r = \%?a;\ M^f$
$M^b = \%!a;\ M^r$



Not possible!
s≠k



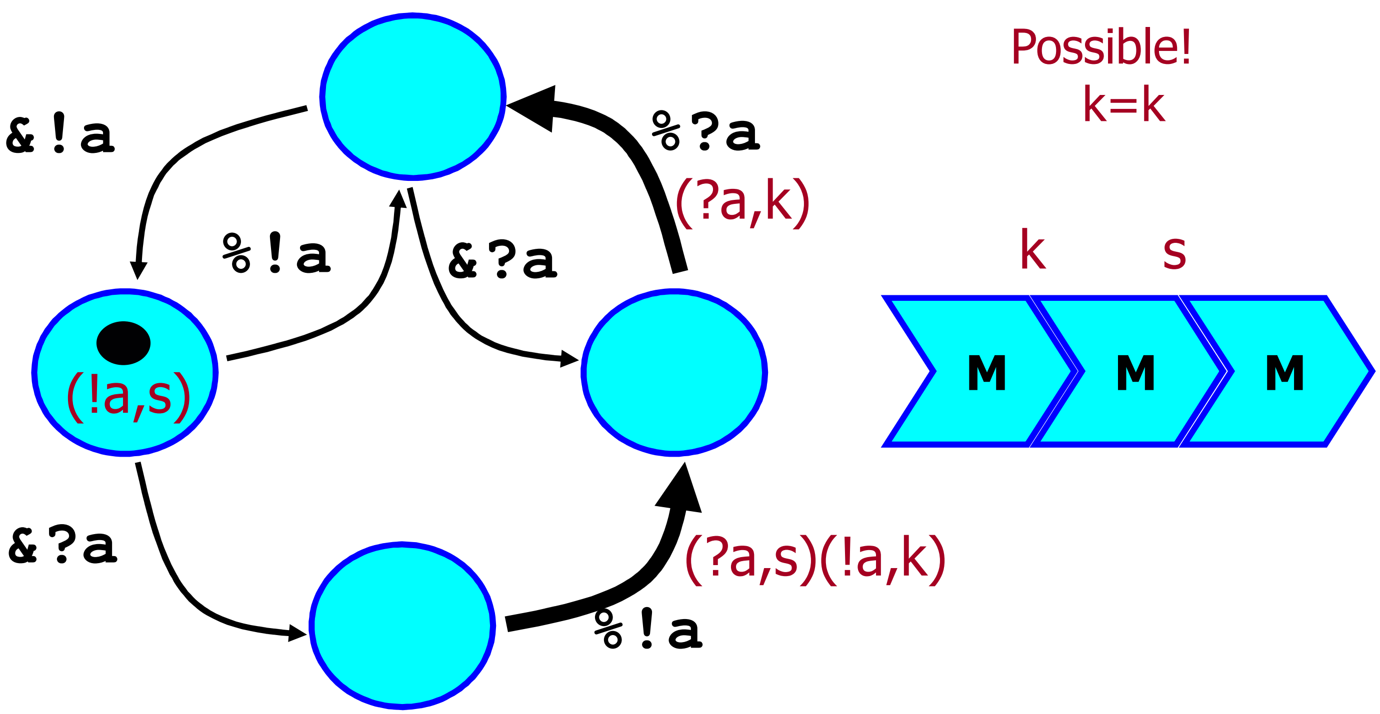
Ex 2: Actin Polymerization

$M^f = \&!a;\ M^l \oplus \&?a;\ M^r$

$M^l = \%!a;\ M^f \oplus \&?a;\ M^b$

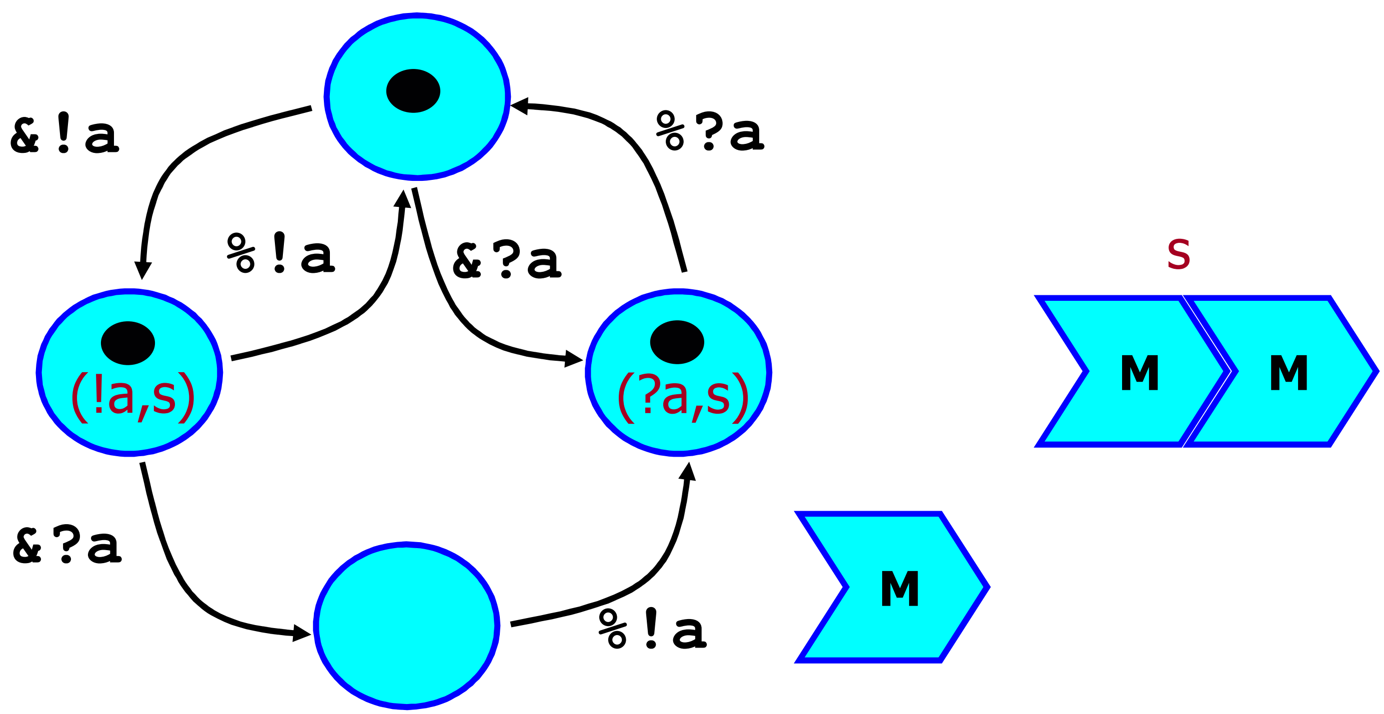
$M^r = \%?a;\ M^f$

$M^b = \%!a;\ M^r$



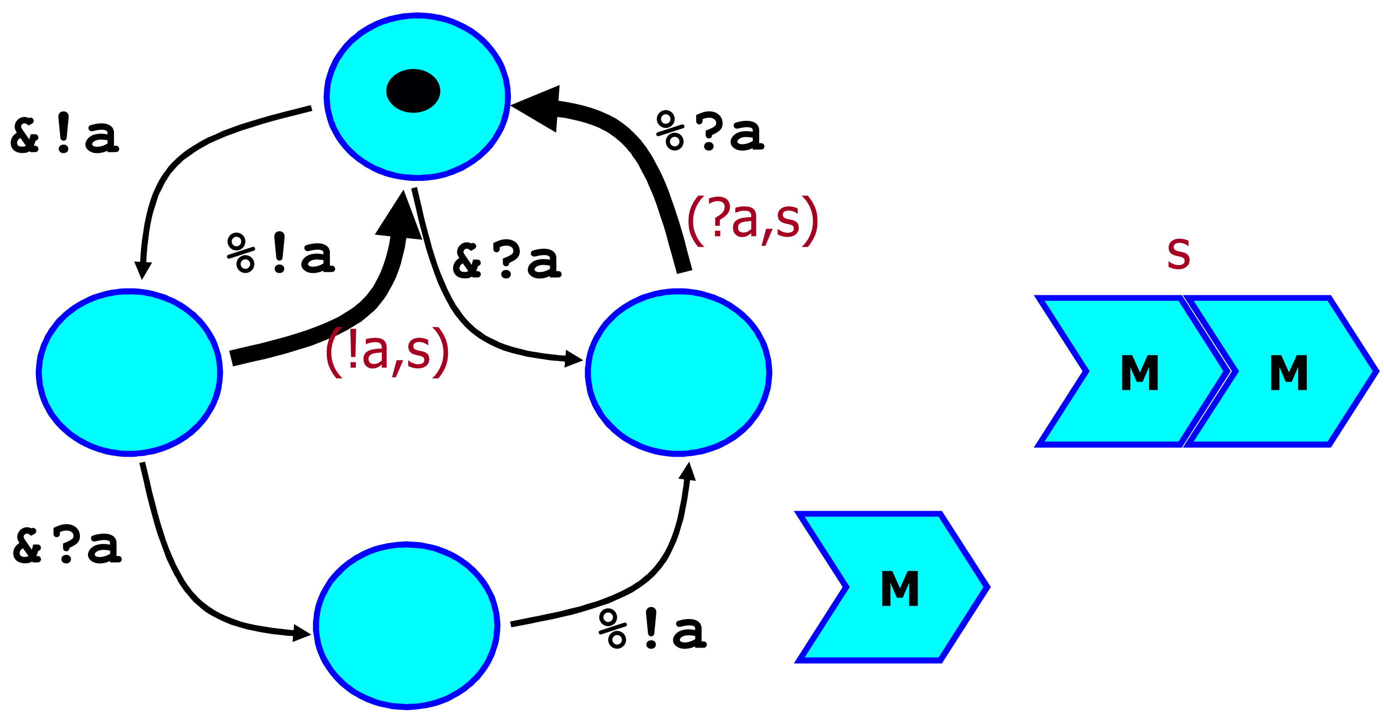
Ex 2: Actin Polymerization

$M^f = \&!a; M^l \oplus \&?a; M^r$
$M^l = \%!a; M^f \oplus \&?a; M^b$
$M^r = \%?a; M^f$
$M^b = \%!a; M^r$



Ex 2: Actin Polymerization

M^f = &!a; M^l ⊕ &?a; M^r
M^l = %!a; M^f ⊕ &?a; M^b
M^r = %?a; M^f
M^b = %!a; M^r



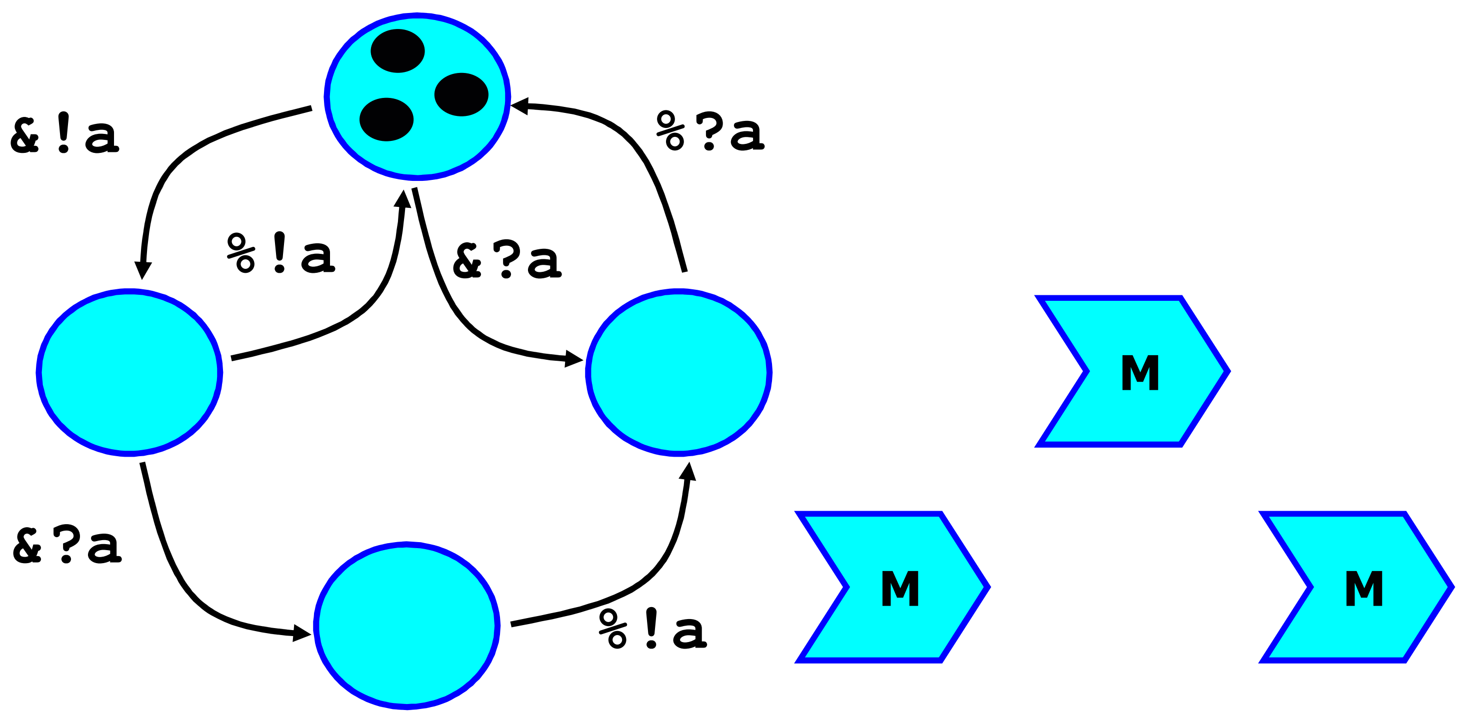
Ex 2: Actin Polymerization

$M^f = \&!a;\ M^l \oplus \&?a;\ M^r$

$M^l = \%!a;\ M^f \oplus \&?a;\ M^b$

$M^r = \%?a;\ M^f$

$M^b = \%!a;\ M^r$



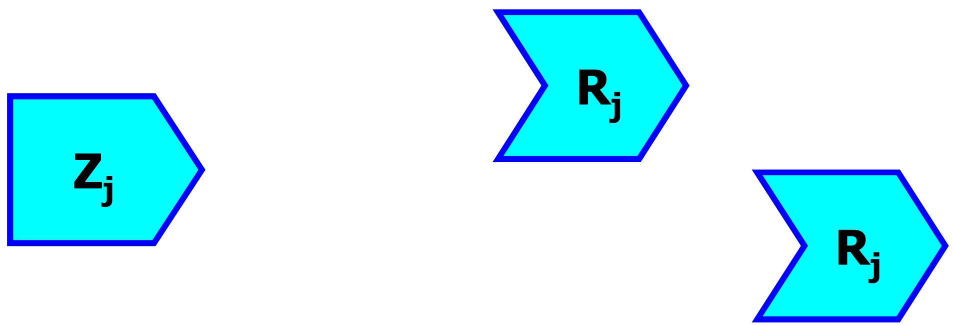
Turing completeness of BGF

- Random Access Machines: [Min67]
  - **Registers**: $r_1 \dots r_n$ hold natural numbers
  - **Program**: sequence of numbered instructions
    - **i: Inc(r_j)**: add 1 to the content of r_j and go to the next instruction
    - **i: DecJump(r_j,s)**: if the content of r_j is not 0 then decrease by 1 and go to the next instruction; otherwise jump to instruction s

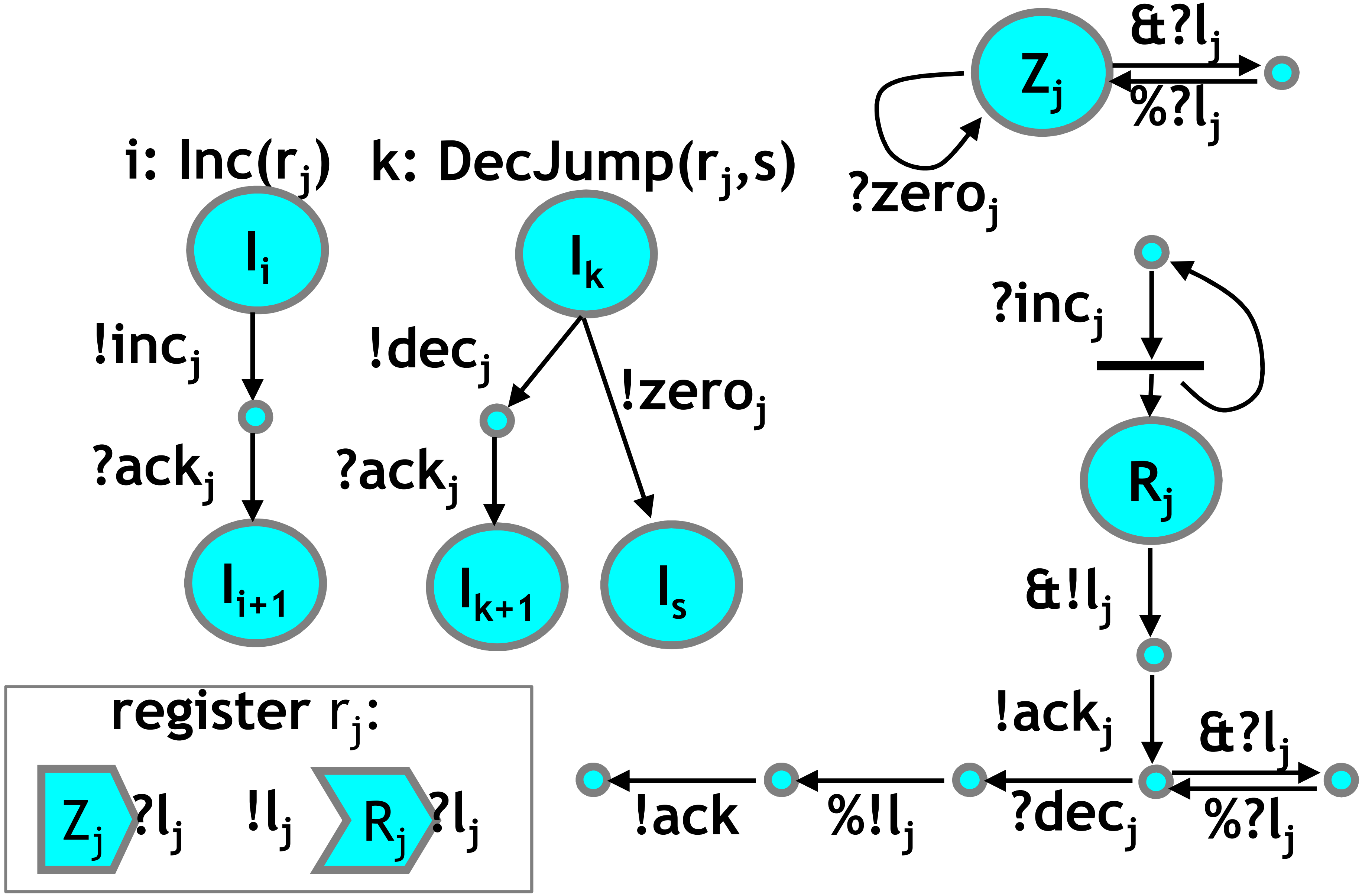
- There is a RAM encoding in BGF
  - But not, as we already showed, in CGF.

Registers as Polymers

- Initially empty register r_j: a seed $\mathbf{Z_j}$
- Increment on r_j: produce a new monomer and associate it to the polymer
- Decrement on r_j: remove last monomer



RAM encoding in BGF



Termination Problems in Chemical Kinetics

Probability Measure for a Markov Chain

- 1-step probability
  - If a state A has n outgoing transitions to states $B_1, \ldots, B_n$, labeled with rates $r_1, \ldots, r_n$, the probability of going from A to B_k in one step is:
  - $p^{(1)}(A,B_k) = r_k / \Sigma_i r_i$

- Many-step probability (Chapman-Kolmogorov equation)
  - The probability of going from A to B in n+m steps is the sum of all ways of going in n steps form A to any X and then in m steps from X to B.
  - $p^{(n+m)}(A,B) = \Sigma_X p^{(n)}(A,X)\, p^{(m)}(X,B)$

Termination Problems

- Probability Measure
  - Let p be the probability measure associated to the computations in a CGF (E,P) that lead to a terminated solution.

- Existential Termination
  - (E,P) existentially terminates if $p > 0$.

- Universal Termination
  - (E,P) universally terminates if $p = 1$.

- Probabilistic Termination
  - (E,P) terminates with probability higher than $0 < \varepsilon < 1$, if $p > \varepsilon$.

Termination Results

	Stochastic	Nondeterministic
Existential Termination	Decidable[1]	Decidable[4]
Universal Termination	Undecidable[2]	Decidable[5]
Probabilistic Termination	Undecidable[3]	N.A.

- Chemical kinetics is not Turing-complete[1]
- Chemical kinetics is Turing-complete up to an arbitrary error[3]
- Existential Termination is equally hard in stochastic and nondeterministic[1,4]
- Universal termination is harder in stochastic than in nondeterministicc[2,5]
- The fairness implicit in stochastic computation makes checking universal termination undecidable[2]

([1,3] due to Soloveichik et. al., Natural Computing 2008)

Conclusions

- Chemistry (CGF) is not Turing complete
  - It is decidable weather given a molecule will be produced.
  - Surprisingly (since this is decidable nondeterministically), it is undecidable whether a program will terminate with probability measure 1.
  - However, chemistry can (slowly) approximate a Turing machine to any degree of precision: it is undecidable whether a given molecule is *likely* to be produced.

- Biochemistry (BGF) is Turing complete.
  - Of course, π-calculus is Turing complete too, but it contains operators that do not have a direct biological interpretation.
  - The BGF a minimal extension of chemistry with biologically inspired operators (complexation/decomplexation) and is already Turing complete
  - Finite Turing-powerful programming constructs can be found in biochemistry but not in basic chemistry.